

Zur Anwendung neuronaler Netzwerke in adaptiven Flugregelungssystemen

Von der Fakultät für Maschinenbau
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung der Würde

eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation

von: Thomas Krüger

aus: Lübeck

eingereicht am: 13.03.2012

mündliche Prüfung am: 24.05.2012

Referenten: Prof. Dr.-Ing. Peter Vörsmann
Prof. Dr.-Ing. Robert Luckner
Prof. Dr.-Ing. habil. Joachim K. Axmann

Vorsitzender: Prof. Dr.-Ing. habil. Georg-Peter Ostermeyer

2012

Vorwort

Die vorliegende Arbeit entstand in den Jahren 2007 bis 2012 während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Luft- und Raumfahrtssysteme der Technischen Universität Braunschweig.

An dieser Stelle möchte ich mich beim Institutsleiter Prof. Dr.-Ing. Peter Vörsmann für die mir im Rahmen meiner Forschungstätigkeit eingeräumten Freiheiten und bei Prof. Dr.-Ing. habil. Joachim Axmann für das stetige Interesse an dieser Arbeit sowie die hilfreichen Ratschläge bedanken. Ebenso danke ich Prof. Dr.-Ing. Robert Luckner für die Übernahme des Korreferates und Prof. Dr.-Ing. habil. Georg-Peter Ostermeyer für die Führung des Prüfungsvorsitzes.

Gleichzeitig möchte ich hervorheben, dass das gute Gelingen dieser Arbeit nicht ohne die stets kollegiale wie angenehme Atmosphäre am Institut und in der Arbeitsgruppe *Unbemannte Flugsysteme* möglich gewesen wäre. Weiteren Dank schulde ich den Herren Dr. Andreas Kuhn und Toni Palau von der Firma Andata, die mit ihrer Unterstützung sowie den gemeinsamen fachlichen Diskussionen viel zum Erfolg der vorliegenden Arbeit beigetragen haben.

Bei meinen Kollegen Arne Krüger und Philipp Schnetter bedanke ich mich für die hilfreichen, kritischen Diskussionen und nicht zuletzt auch bei Herrn Thomas Schmidtke für die dauerhafte Unterstützung in der Endphase dieser Arbeit.

Ganz besonderer Dank gilt meiner Familie - allen voran meiner Frau Susanne, meiner Mutter Brigitte sowie meinen Brüdern Kay und Axel. Ihre geduldige wie dauerhafte Unterstützung sowohl während meiner Ausbildungszeit als auch beim Erstellen dieser Arbeit war von unschätzbarem Wert.

Braunschweig im August 2012

Thomas Krüger

Übersicht

Die vorliegende Arbeit stellt einen Beitrag zur systematischen Einbindung künstlicher neuronaler Netzwerke in adaptive Regelungssysteme dar. Neben den zentralen Grundlagen zur Abbildung nichtlinearer dynamischer Prozesse mit neuronalen Netzen wird hierzu eine strukturierte Methodik des Netzwerkentwurfes dargestellt, die eine statistische Bewertung des Lernerfolges ermöglicht.

Darüber hinaus wird im Hinblick einer verbesserten Transparenz zur Regelung genutzter neuronaler Netzwerke ein Paradigmenwechsel bei der Betrachtung ihrer Lernverfahren vollzogen. Demnach werden neuronale Netze als dynamische Systeme aufgefasst, womit das Lernen innerhalb eines Netzwerkes als geregelter Prozess betrachtet werden kann. Damit ist eine Übertragung des Konzeptes der Gleitzustandsregelung auf neuronale Lernverfahren möglich, welche so um eine regelungstechnisch motivierte, intrinsische Stabilitätsbedingung erweitert werden können. Durch die dynamische Bestimmung der Lernraten vermag dieser Lernansatz die Zustände eines neuronalen Netzes in den global asymptotisch stabilen Gleitzustand zu überführen.

Die praktische Bedeutung dieses Lernverfahrens wird am Beispiel zweier stark unterschiedlicher Regelungsstrategien dargestellt: Der sogenannten modellfreien neuronalen Regelung, welche gänzlich ohne mathematisches Modell des geregelten Systems auskommt, und der neuronal gestützten dynamischen Inversion, einem Regelungsansatz, der ein hohes Maß an Kenntnis der Systemdynamik erfordert. Die Ergebnisse beider Regelungsverfahren unterstreichen die fundierten wie robusten Eigenschaften neuronaler Netze zur Abbildung komplexer nichtlinearer dynamischer Prozesse.

Darüber hinaus zeigen sich besondere Vorteile bei der Kombination von dynamischer Inversion und neuronalen Netzwerken, die mit dem Gleitzustandslernverfahren trainiert werden. Dies liegt darin begründet, dass die Überführung der Zustandsgrößen des Netzes in den Gleitzustand eine erweiterte stabilisierende Wirkung auf die Fehlerdynamik des geschlossenen Regelkreises besitzt. Dazu wird am Beispiel eines beschädigten unbemannten Flugzeuges unter Störeinfluss gezeigt, dass das Gleitzustandsverfahren wegen seiner stabilitätsbedingten dynamischen Bestimmung der Lernrate eine höhere Konvergenzgeschwindigkeit als vergleichbare etablierte Ansätze aufweist. Die Ergebnisse unterstreichen in besonderer Weise die Vorzüge der interdisziplinären Kombination von klassischer Regelungstechnik und Maschinenlernverfahren bei der Auslegung komplexer Regelungssysteme.

Abstract

The present work makes a contribution to the systematic integration of artificial neural networks into adaptive control systems. Besides the basic principles of mapping non-linear dynamic processes using neural networks, a structured methodology for the network design is illustrated, which allows a statistical evaluation of the learning success.

In addition, to allow for an improved transparency of neural networks used in control systems a paradigm change regarding their learning methods is undertaken. This means that a neural network is treated as a dynamic system whereby the learning within such a network becomes a controlled process. Therefore, the application of the concept of sliding mode control to learning techniques of neural networks is possible, which thereby can be augmented with an intrinsic stability condition. Through the dynamic calculation of the learning rate this learning approach is able to constrain the network's states into the globally asymptotically stable sliding mode.

The practical relevance of this learning method is illustrated utilising two dissimilar control strategies: the so-called indirect adaptive neuro-control, which does not need any mathematical model of the controlled system, and the nonlinear dynamic inversion, for which a high degree of knowledge of the system dynamics is essential. The results of both control methods underline the profound as well as robust capabilities of neural networks to map complex non-linear dynamic processes.

Furthermore, particular advantages arise from the combination of dynamic inversion and neural networks trained with the sliding mode learning method. This is the case because constraining the network's states to the sliding mode offers an enhanced stabilising effect upon the error dynamics of the closed-loop control system. Using the example of a damaged unmanned aircraft under perturbation it is shown, that the sliding mode learning technique due to the dynamic calculation of the learning rate offers a higher speed of convergence than comparable established methods. The results especially emphasise the advantages of the interdisciplinary combination of control theory and machine learning techniques for the design of complex control systems.

Inhaltsverzeichnis

| | |
|---|-------------|
| Abbildungsverzeichnis | XIII |
| Tabellenverzeichnis | XVII |
| Formelzeichen | XIX |
| 1 Einleitung | 1 |
| 1.1 Stand der Technik | 2 |
| 1.2 Gegenstand der Arbeit | 4 |
| 1.3 Gliederung der Ausarbeitung | 5 |
| 2 Grundlagen neuronaler Netzwerke | 7 |
| 2.1 Biologisches Vorbild und technisches Neuronenmodell | 7 |
| 2.1.1 Transferfunktionen | 9 |
| 2.2 Topologien neuronaler Netzwerke | 10 |
| 2.3 Lernverfahren | 12 |
| 2.3.1 Lernen durch Rückpropagation des Fehlers | 13 |
| 2.3.2 Gradientenabstieg mit zusätzlichem Moment | 15 |
| 2.3.3 Der Levenberg-Marquardt-Algorithmus | 16 |
| 2.4 Universelle Approximationseigenschaften neuronaler Netze | 17 |
| 2.5 Abbildung dynamischer Prozesse mit neuronalen Netzen | 18 |
| 2.6 Anwendungsarten neuronaler Lernverfahren | 19 |
| 2.6.1 Trainingsdatengewinnung und -aufbereitung | 21 |
| 2.6.2 Bewertung der Generalisierungsfähigkeit | 22 |
| 2.7 Systematischer Entwurfsprozess für neuronale Netze | 23 |
| 3 Anwendung der strukturvariablen Regelung auf neuronale Lernverfahren | 27 |
| 3.1 Grundbegriffe der strukturvariablen Regelung | 28 |
| 3.2 Stabilitätsbetrachtung nichtlinearer Systeme | 31 |
| 3.3 Systeme im Gleitzustand | 33 |
| 3.3.1 Existenz und Erreichbarkeit des Gleitzustandes | 34 |
| 3.3.2 Invarianz und Robustheit | 36 |
| 3.4 Anwendung der Gleitzustandsregelung auf neuronale Netzwerke | 37 |
| 3.4.1 Definition der Schaltfunktion | 38 |

| | | |
|----------|---|------------|
| 3.4.2 | Herleitung des Stellgesetzes | 40 |
| 3.5 | Darstellung neuronaler Netzwerke im Gleitzustand | 44 |
| 4 | Flugsimulation und Reglerimplementierung | 49 |
| 4.1 | Nichtlineare Zustandsgleichungen der Flugzeugbewegung | 50 |
| 4.2 | Modellannahmen und Umsetzung der Regelungsverfahren | 52 |
| 5 | Modellfreie neuronale Regelung | 55 |
| 5.1 | Lernen innerhalb eines Regelkreises | 56 |
| 5.2 | Aufbau des modellfreien neuronalen Flugreglers | 57 |
| 5.2.1 | Wahl der Regelgröße zur Ermittlung des Lernsignals | 60 |
| 5.2.2 | Auslegung des neuronalen Reglers | 62 |
| 5.2.3 | Auslegung des neuronalen Beobachters | 64 |
| 5.2.4 | Beschreibung des Referenzmodells | 66 |
| 5.3 | Erweiterung des Regelkreises | 67 |
| 5.3.1 | Stabilitätsbetrachtung nach Ljapunow | 69 |
| 5.4 | Implementierung des Vorwissens | 71 |
| 5.4.1 | Bewertung des statistischen Trainings | 73 |
| 5.4.2 | Bewertung des Reglertrainings | 75 |
| 5.4.3 | Bewertung des Beobachtertrainings | 77 |
| 6 | Nichtlineare dynamische Inversion | 79 |
| 6.1 | Grundprinzip der dynamischen Inversion | 80 |
| 6.1.1 | Nichtlineare Zustandstransformation | 82 |
| 6.1.2 | Linearisierende Zustandsrückführung | 84 |
| 6.1.3 | Interne Dynamik des Regelkreises | 86 |
| 6.1.4 | Referenzmodelle | 87 |
| 6.1.5 | Fehlerdynamik und linearer Regler | 89 |
| 6.2 | Dynamische Beschränkung der Ersatzregelgrößen | 93 |
| 6.3 | Erweiterung durch neuronale Netzwerke | 96 |
| 6.3.1 | Fehlerdynamik der erweiterten Reglerarchitektur | 98 |
| 6.3.2 | Stabilitätsbetrachtung des erweiterten Systems | 103 |
| 6.3.3 | Erweiterung mit dem Gleitzustandslernverfahren | 105 |
| 6.4 | Umsetzung der Regelungsstrategie | 109 |
| 6.4.1 | Anwendung auf die Rotationsdynamik | 111 |
| 6.4.2 | Anwendung auf die Lageregelung | 115 |
| 6.4.3 | Anwendung auf die Bahndynamik und Bahnvorgabe | 118 |
| 6.4.4 | Steuerflächenzuweisung | 121 |
| 6.5 | Parameterwahl für das Gesamtsystem | 122 |
| 7 | Ausgewählte Simulationsergebnisse | 127 |
| 7.1 | Vergleich der Regelungsansätze | 128 |

| | | |
|----------|---|------------|
| 7.2 | Einfluss der Lernverfahren auf die dynamische Inversion | 135 |
| 8 | Zusammenfassung und Ausblick | 151 |
| 8.1 | Zukünftige Forschungsansätze | 155 |
| | Literaturverzeichnis | 157 |
| A | Anhang | 167 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 1.1 | Die unbemannten Flugsysteme <i>Carolo P200</i> und <i>Carolo P360</i> | 4 |
| 2.1 | Darstellung eines künstlichen Neurons. | 9 |
| 2.2 | Typische Transferfunktionen neuronaler Netze. | 9 |
| 2.3 | Darstellung eines dreischichtigen Netzes mit einem Ausgabeneuron. | 11 |
| 2.4 | Abbildung eines dynamischen Prozesses durch ein neuronales Netzwerk. . . . | 19 |
| 2.5 | Bewertung der Generalisierungsfähigkeit anhand des Validationsfehlers. . . . | 23 |
| 2.6 | Systematischer Entwurf und Bewertung neuronaler Netzwerke. | 24 |
| 3.1 | Prinzipielle Darstellung der Strukturumschaltung zwischen zwei Reglern. . . . | 29 |
| 3.2 | Erreichen des Gleitzustandes bei einem System zweiter Ordnung. | 30 |
| 3.3 | Asymptotische Stabilität am Beispiel von Höhenlinien einer Ljapunow-Funktion. | 32 |
| 3.4 | Das Training eines neuronalen Netzes als Regelungsprozess. | 37 |
| 3.5 | Zielfunktion und Antworten bei unterschiedlichem λ sowie Eingangssignale. . | 45 |
| 3.6 | Phasendiagramm und Lernraten bei unterschiedlicher Schaltfunktion. | 46 |
| 3.7 | Netzantworten und Phasendiagramm bei einem Sprung der Zielgröße. | 47 |
| 3.8 | Netzantworten bei einem Sprung der Zielgröße und zusätzlichem Rauschen. . . | 48 |
| 4.1 | Übersichtsdarstellung der nichtlinearen Flugsimulation. | 50 |
| 5.1 | Allgemeine Darstellung eines lernfähigen Flugregelkreises mit Kritikelement. . | 56 |
| 5.2 | Darstellung des modellfreien neuronalen Flugregelkreises. | 58 |
| 5.3 | Darstellung einer kubischen Spline-Kurve. | 61 |
| 5.4 | Verschaltung des neuronalen Reglers. | 63 |
| 5.5 | Verschaltung des neuronalen Beobachters. | 65 |
| 5.6 | Lineares Referenzmodell zweiter Ordnung. | 67 |
| 5.7 | Darstellung des erweiterten modellfreien neuronalen Flugregelkreises. | 68 |
| 5.8 | Darstellung des erweiterten Referenzmodells. | 69 |
| 5.9 | Profil der Referenzflugbahn in Längs- und Seitenbewegung. | 72 |
| 5.10 | Verteilungsfunktion des Fehlers aller Reglernetzwerke. | 73 |
| 5.11 | Verlauf des Fehlers über der Anzahl verdeckter Neuronen aller Reglernetze. . | 74 |
| 5.12 | Antwortverhalten eines Reglernetzwerkes für die Referenzflugbahn. | 75 |
| 5.13 | Streudiagramm von Zielwerten und Netzantworten eines Reglernetzwerkes. . . | 76 |
| 5.14 | Antwortverhalten eines Reglernetzwerkes für die Referenzflugbahn. | 77 |

| | | |
|------|--|-----|
| 6.1 | Vereinfachte Darstellung eines Regelkreises auf Basis dynamischer Inversion. | 79 |
| 6.2 | Darstellung der linearisierenden Zustandsrückführung. | 86 |
| 6.3 | Darstellung des vereinfachten Ersatzsystems. | 89 |
| 6.4 | Darstellung des durch einen linearen Regler erweiterten Regelkreises. | 92 |
| 6.5 | Darstellung des vorgesteuerten Referenzmodells im Regelkreis. | 94 |
| 6.6 | Durch Vorsteuerung beeinflusstes Referenzmodell zweiter Ordnung. | 95 |
| 6.7 | Darstellung des durch ein neuronales Netzwerk erweiterten Regelkreises. | 99 |
| 6.8 | Übersicht des geschlossenen Regelkreises und aller seiner Erweiterungen. | 110 |
| 6.9 | Anwendung der Regelung auf das Mehrgrößensystem der Rotationsdynamik. | 113 |
| 6.10 | Um Begrenzungen erweitertes Referenzmodell erster Ordnung. | 118 |
| 7.1 | Komponenten des Windvektors überlagert mit einem Turbulenzspektrum. | 128 |
| 7.2 | Nicht maßstäbliches Profil der Testflugbahn in Längs- und Seitenbewegung. | 129 |
| 7.3 | Seitliche Bahnabweichung einer unbekannten Bahn unter Störeinfluss. | 130 |
| 7.4 | Ausgabe und Ausgabefehler des neuronalen Beobachters. | 130 |
| 7.5 | Seitliche Bahnabweichung bei erhöhtem Wind mit $u_{Wg} = 5$ m/s. | 131 |
| 7.6 | Durch Störungen angeregte Schwingung des Hängewinkels. | 132 |
| 7.7 | Profil einer wissenschaftlichen Flugmission in Längs- und Seitenbewegung. | 132 |
| 7.8 | Seitliche Bahnabweichung für die exemplarische Flugmission. | 133 |
| 7.9 | Seitliche Bahnabweichung bei unterschiedlichen Windverhältnissen. | 134 |
| 7.10 | Darstellung des Hängewinkels bei erhöhtem Wind mit $u_{Wg} = 5$ m/s. | 134 |
| 7.11 | Darstellung des tatsächlichen und gemessenen Windes. | 135 |
| 7.12 | Größen der Reglerkaskade der Längsbewegung unter Windeinfluss. | 137 |
| 7.13 | Größen der Reglerkaskade der Seitenbewegung unter Windeinfluss. | 138 |
| 7.14 | Sprung des Inversionsfehlers in der Längsbewegung bei idealem Flugzeug. | 139 |
| 7.15 | Sprung des Inversionsfehlers in der Seitenbewegung bei idealem Flugzeug. | 140 |
| 7.16 | Sprung des Inversionsfehlers in der Längsbewegung bei Unsicherheiten. | 141 |
| 7.17 | Sprungantwort von Lernrate und Gewichtsnorm. | 142 |
| 7.18 | Destabilisierung des Netzwerkes bei zu hoher fester Lernrate. | 143 |
| 7.19 | Größen der Reglerkaskade der Längsbewegung mit Schadensfall. | 146 |
| 7.20 | Größen der Reglerkaskade der Seitenbewegung mit Schadensfall. | 147 |
| 7.21 | Ausgabe des Netzwerkes zur Stützung der Nickgeschwindigkeit. | 148 |
| 7.22 | Vergleich der Höhenabweichung der verschiedenen Lernverfahren. | 149 |
| A.1 | Seitliche Bahnabweichung der Beispielmission mit dynamischer Inversion. | 167 |
| A.2 | Hängewinkelverlauf des <i>P360</i> unter Einfluss atmosphärischer Störungen. | 167 |
| A.3 | Größen der Längsbewegung unter Windeinfluss mit Modellfehler von -10 %. | 168 |
| A.4 | Größen der Seitenbewegung unter Windeinfluss mit Modellfehler von -10 %. | 169 |
| A.5 | Größen der Längsbewegung unter Windeinfluss mit Modellfehler von +10 %. | 170 |
| A.6 | Größen der Seitenbewegung unter Windeinfluss mit Modellfehler von +10 %. | 171 |
| A.7 | Vergleich der seitlichen Bahnabweichung der verschiedenen Lernverfahren. | 172 |
| A.8 | Höhenfehler des <i>P360</i> beim Flug mit Systemschaden. | 172 |

| | | |
|-----|---|-----|
| A.9 | Seitliche Bahnabweichung des <i>P360</i> beim Flug mit Systemschaden. | 172 |
|-----|---|-----|

Tabellenverzeichnis

| | | |
|-----|---|-----|
| 3.1 | Vergleich der Fehlerwerte bei verschiedenen Rauschintensitäten. | 48 |
| 5.1 | Zusammensetzung des Vektors \vec{x} | 64 |
| 5.2 | Fehlerbetrachtung des neuronalen Reglers. | 77 |
| 5.3 | Fehlerbetrachtung des neuronalen Beobachters. | 78 |
| 6.1 | Mögliche Eingangssignale der Längsbewegung. | 114 |
| 6.2 | Mögliche Eingangssignale der Seitenbewegung. | 115 |
| 7.1 | Technische Daten der unbemannten Flugsysteme Carolo <i>P200</i> und <i>P360</i> | 127 |

Formelzeichen

| Lateinische Buchstaben | | Einheiten |
|----------------------------|--|------------------|
| $\mathbf{A}(\vec{x})$ | Entkopplungsmatrix | — |
| \mathbf{A} | Koeffizientenmatrix des erweiterten Referenzmodells | — |
| \mathbf{A} | Übertragungsmatrix eines dynamischen Systems | div. |
| \mathbf{A}_E | Systemmatrix der Fehlerdynamik | div. |
| A | Auftrieb | N |
| $a(\vec{\xi}, \vec{\eta})$ | Element der Entkopplungsmatrix \mathbf{A} | div. |
| a_i | Koeffizienten eines charakteristischen Polynoms | — |
| a_i | Substitutionsparameter der Lernratenberechnung | — |
| a_m | m-ter x-Koeffizient einer Spline-Kurve | — |
| a_y | Querbeschleunigung | m/s ² |
| \mathbf{B} | Eingangsmatrix eines dynamischen Systems | div. |
| \mathbf{B} | Koeffizientenmatrix des erweiterten Referenzmodells | — |
| $\underline{\mathbf{B}}$ | Transformationsvorschrift innerer Störungen | — |
| \mathbf{b}_E | Eingangsmatrix der Fehlerdynamik | div. |
| \vec{b} | Vektor der Ableitungen von \vec{h} entlang \vec{f} | div. |
| b | Spannweite | m |
| b_i | Substitutionsparameter der Lernratenberechnung | — |
| b_m | m-ter y-Koeffizient einer Spline-Kurve | — |
| C | aerodynamischer Beiwert, aerodynamisches Derivat | — |
| c_i | Rückführkoeffizienten der linearen Fehlerdynamik | div. |
| D | Dämpfungsgrad | — |
| D_i | Hauptabschnittsdeterminanten einer Hurwitz-Matrix | div. |
| d | seitliche Bahnabweichung, <i>cross track error</i> | m |

| | | |
|-------------------------|--|------------------------|
| E | Einheitsmatrix | — |
| E | quadratische Fehlerfunktion eines neuronalen Netzwerkes | — |
| \vec{e}, e | (vektorieller) Regelfehler | div. |
| $\hat{\mathbf{F}}$ | Übertragungsfunktion des Systemmodells | div. |
| $F(s)$ | Übertragungsfunktion eines linearen Systems | div. |
| \vec{F} | Kraftvektor | N |
| \vec{F}_{TW} | Triebwerksschub | N |
| \vec{f} | Transformationsvorschrift innerer Störungen | — |
| \vec{f}_s | Vektorfeld der äußeren Störungen | div. |
| $\underline{\vec{f}}_s$ | Transformationsvorschrift äußerer Störungen | — |
| f_i | Übertragungsfunktion eines Neurons i | — |
| G | Übertragungsmatrix der nichtlinearen Stelldynamik | div. |
| $G(s)$ | Übertragungsfunktion | div. |
| \vec{G} | Gewichtskraft | N |
| \vec{g} | nichtlineare Stelldynamik eines Eingrößensystems | div. |
| g | Erdbeschleunigung | m/s ² |
| H | Hesse-Matrix | — |
| H | Hurwitz-Matrix | div. |
| H | Flughöhe | m |
| \vec{h} | vektorielle Ausgangsfunktion eines nichtlinearen Systems | div. |
| J | Jacobi-Matrix | — |
| K | Parametermatrix des <i>reaching law</i> | — |
| K | Matrix der Reglerverstärkungen | — |
| K | Verstärkungsfaktor | — |
| k | Formfaktor der Datennormalisierung | — |
| k_{r0} | Koeffizient des robustifizierenden Terms | — |
| k_{r1} | Koeffizient des robustifizierenden Terms | — |
| \vec{L} | Drehimpulsvektor | kg · m ² /s |
| L | Lie-Ableitung | — |
| L | Rollmoment | Nm |
| \mathbf{M}_{ab} | Transformationsmatrix von System b zu a | ◦ |
| \vec{M} | Momentenvektor | Nm |

| | | |
|----------------------------------|---|---------------------|
| M | Nickmoment | Nm |
| m | Masse | kg |
| N | Giermoment | Nm |
| n | Lastvielfaches | — |
| n | Ordnung eines dynamischen Systems | — |
| n_i | Aktivierungszustand eines Neurons i | — |
| O | Restterm einer Taylorreihenentwicklung | — |
| o_i | Ausgabe eines Neurons i | — |
| \mathbf{P}_E | positive, symmetrische Matrix der linearen Ljapunow-Gleichung | div. |
| P_m | m-ter Punkt einer Spline-Kurve | — |
| $\vec{p}(\vec{\xi}, \vec{\eta})$ | von \vec{u} abhängiger Teil der internen Dynamik | div. |
| \vec{p} | Vektor der Störparameter | — |
| p | Rollrate | rad/s |
| \mathbf{Q} | Parametermatrix des <i>reaching law</i> | — |
| \mathbf{Q}_E | positive, symmetrische Matrix der linearen Ljapunow-Gleichung | div. |
| Q | Querkraft | N |
| $\vec{q}(\vec{\xi}, \vec{\eta})$ | von \vec{u} unabhängiger Teil der internen Dynamik | div. |
| q | Nickrate | rad/s |
| \bar{q} | Staudruck | N/m ² |
| \vec{R}^A | Vektor der aerodynamischen Kräfte | N |
| \vec{r}, r | (vektorieller) relativer Grad eines Systems | — |
| r | Gierrate | rad/s |
| \vec{S}, S | (vektorielle) Schaltfunktion | — |
| S | Bezugsflügelfläche | m ² |
| \vec{s} | Positionsvektor | m |
| s | Laplace-Variable | — |
| \mathbf{T} | Trägheitstensor | kg · m ² |
| T | Zeitkonstante eines linearen Systems erster Ordnung | s |
| T_s | zeitliche Schrittweite im Regelkreis | — |
| \vec{u}, u | (vektorielle) Eingangsgröße eines dynamischen Systems | div. |
| u | x-Komponente einer indizierten Geschwindigkeit | m/s |
| \vec{V}_K | Vektor der Bahngeschwindigkeit | m/s |

| | | |
|------------------------|---|------|
| V | Ljapunow-Funktion | — |
| v | y-Komponente einer indizierten Geschwindigkeit | m/s |
| W | Widerstand | N |
| \mathbf{w} | Gesamtgewichtsmatrix eines Netzwerkes | — |
| $\mathbf{w}^{(1)}$ | Gewichtsmatrix der Schicht 1 | — |
| $\mathbf{w}^{(2)}$ | Gewichtsmatrix der Schicht 2 | — |
| $\Delta \mathbf{w}$ | Gewichtsänderungen | — |
| w | z-Komponente einer indizierten Geschwindigkeit | m/s |
| w_{ij} | Gewicht zwischen den Neuronen i und j | — |
| \vec{x} | Zustandsvektor eines dynamischen Systems | — |
| $\underline{\vec{x}}$ | Eingangsvektor eines neuronalen Netzwerkes | div. |
| x | geodätische x-Koordinate | m |
| \vec{y}, y | (vektorielle) Ausgangsgröße eines dynamischen Systems | div. |
| \vec{y}, y | (vektorielle) Ausgabe eines neuronalen Netzwerkes | div. |
| $\Delta \vec{y}_t$ | Änderung der Netzausgabe | div. |
| $\Delta \vec{y}_{k,t}$ | Änderung der kommandierten Netzausgabe | div. |
| y | geodätische y-Koordinate | m |
| \mathbf{z} | Störterm bei der Netzwerklinearisierung | — |
| \vec{z} | Zustandsvektor des transformierten Systems | div. |
| z | geodätische z-Koordinate | m |

Griechische Buchstaben

| | | |
|-------------------------|--|------|
| $\vec{\alpha}(\vec{x})$ | Koeffizientenvektor der linearisierenden Zustandsrückführung | div. |
| α | Anstellwinkel | ° |
| α_K | Bahnanstellwinkel | ° |
| $\vec{\beta}(\vec{x})$ | Koeffizientenvektor der linearisierenden Zustandsrückführung | div. |
| β | Schiebewinkel | ° |
| β_K | Bahnschiebewinkel | ° |
| $\vec{\chi}$ | Vektor der Regelfehler und ihrer Ableitungen | div. |
| χ | Bahnazimut | ° |
| $\vec{\delta}$ | Vektor der Ausgangsempfindlichkeit einer Schicht | — |
| $\vec{\delta}$ | Vektor der rückpropagierten Fehler einer Schicht | — |

| | | |
|------------------------|---|-------|
| δ_j | rückpropagierter Fehler eines Neurons j | — |
| $\vec{\Delta}, \Delta$ | (vektorieller) Inversionsfehler | div. |
| $\vec{\epsilon}_A$ | Approximationsfehler eines neuronalen Netzwerkes | div. |
| $\vec{\epsilon}$ | Ausgabefehler eines neuronalen Netzwerkes | div. |
| $\vec{\eta}$ | Teilzustandsvektor der internen Dynamik | div. |
| η | Höhenruderausschlag | ° |
| γ | Bahnwinkel | ° |
| Γ | Matrix der Lernraten | — |
| λ | Koeffizient der e-Modifikation, Faktor des Vergessens | — |
| λ | Überblendfaktor des Levenberg-Marquardt Algorithmus' | — |
| λ_i | Koeffizient einer Schaltfunktion | — |
| μ | Erwartungswert | — |
| μ | Lernrate | — |
| μ_K | Bahnhängewinkel | ° |
| ∇ | Nablaoperator | — |
| \vec{v}, v | (vektorielle) Ersatzregelgröße | div. |
| v_{ad} | adaptiver Anteil der Ersatzregelgröße | div. |
| v_r | robustifizierender Anteil der Ersatzregelgröße | div. |
| Ω | betrachteter Bereich des Zustandsraumes | — |
| $\vec{\omega}$ | Vektor der Drehgeschwindigkeiten | rad/s |
| ω_0 | Eigenfrequenz eines ungedämpften Systems | rad/s |
| Φ | nichtlineare Funktion der Koordinatentransformation | div. |
| Φ | Hängewinkel | ° |
| Ψ | Gierwinkel, Steuerkurs | ° |
| σ | Momentenkoeffizient | — |
| σ | Standardabweichung | — |
| $\vec{\zeta}$ | (vektorielle) gefilterte Fehlergröße | div. |
| Θ | Nickwinkel | ° |
| $\vec{\xi}$ | Teilzustandsvektor der linearisierten Eingangs-/Ausgangsdynamik | div. |
| ξ | Querruderausschlag | ° |
| ζ | Seitenruderausschlag | ° |

Indices

| | |
|----------|--|
| $*$ | Kennzeichnung der optimalen Gewichtsmatrizen |
| 0 | Kennzeichnung des betrachteten Referenzpunktes |
| ω | Kennzeichnung der Rotationsdynamik eines Flugzeuges |
| A | aerodynamische Größen |
| b | Kennzeichnung eines Bias-Gewichtes |
| BN | Beobachternetzwerk |
| e | Kennzeichnung des erweiterten Referenzsignals |
| eq | Kennzeichnung der äquivalenten Regelung |
| F | Kennzeichnung der Frobeniusnorm |
| f | flugzeugfeste Koordinaten |
| g | geodätische Koordinaten |
| h | Kennzeichnung des Vorsteuersignals der Referenzmodelle |
| i | Neuron i einer Schicht l |
| j | Neuron j einer Schicht $l + 1$ |
| K | kinematische Größen |
| k | bahnfeste Koordinaten |
| k | kommandierter Wert, Zielwert |
| l | Schichtindex eines Netzwerkes |
| LR | Kennzeichnung des linearen Reglers der dynamischen Inversion |
| NN | Größe eines neuronalen Netzwerkes |
| q | Anzahl der Neuronen einer Schicht |
| R | Referenzsignal der dynamischen Inversion |
| r | Referenzwert |
| RN | Reglernetzwerk |
| t | Kennzeichnung des aktuellen Zeitschrittes |
| t | Laufparameter auf einer Spline-Kurve |
| \sim | Kennzeichnung der Gewichtsfehler |
| \wedge | Kennzeichnung von Modellgrößen der dynamischen Inversion |

Abkürzungen

| | |
|---------|--|
| DO-178B | Entwicklungsstandard - Software Considerations in Airborne Systems and Equipment Certification |
|---------|--|

| | |
|------|--|
| GZR | Gleitzustandsregelung |
| KI | Künstliche Intelligenz |
| KNN | künstliches neuronales Netzwerk |
| MIMO | <i>multiple input, multiple output</i> - Mehrgrößensystem |
| MSE | <i>mean square error</i> - mittlerer quadratischer Fehler |
| PCH | <i>pseudo control hedging</i> |
| RMSE | <i>root mean square error</i> - Wurzel des mittleren quadratischen Fehlers |
| RP | Rückpropagationsalgorithmus |
| SISO | <i>single input, single output</i> - Eingrößensystem |
| V&V | Verifikation und Validation |

1 Einleitung

Seitdem sich die Regelungstechnik in der Mitte des 20. Jahrhunderts als eine eigenständige wissenschaftliche Disziplin im Ingenieurwesen etabliert hat, ist ihre Anwendung auf technische Systeme, auch auf solche des alltäglichen Gebrauchs, nunmehr selbstverständlich. Dies gilt im besonderen Maße auch für komplexe Regelungsaufgaben wie die der Luft- und Raumfahrt. Bei der Regelung geht es zunächst um die gezielte Veränderung des Verhaltens dynamischer Systeme mit dem Ziel, von außen wirkende Störungen auszugleichen und bestimmte Zielwertvorgaben einzuhalten; diese Aufgaben gehören auch zum Standard heutiger Flugregler. Darüber hinaus zeigen aktuelle Entwicklungen einen stetig wachsenden Automatisierungsgrad komplexer nichtlinearer Systeme, was besonders in der Verkehrstechnik ausgeprägt ist. Die Umsetzung definierter Aufgaben und Prozeduren, also vollautomatisches Verhalten, zum Beispiel für die automatische Landung eines Flugzeuges bei schlechter Witterung oder die Erfüllung einer Mission eines unbemannten Fluggerätes, ist mit den Mitteln der konventionellen linearen Regelungstechnik zuverlässig zu realisieren. Gerade aber bei unbemannten Flugsystemen und nicht zuletzt auch bei Automobilen ist die tatsächliche Autonomie, das selbstständige Eingreifen des Systems in seinen Regelungsprozess, der nächste Entwicklungsschritt.

Wirklich autonomes Verhalten setzt voraus, dass das technische System im Rahmen seines Betriebsumfeldes für eine wahrgenommene Problemstellung selbstständig eine passende Lösung wählt und umsetzt. In diesem Zusammenhang scheint es daher zweckmäßig, die zu Grunde liegende Regelungsstrategie veränderbar, also adaptiv, zu gestalten. Beinhaltet dies auch die Lernfähigkeit, so besitzen adaptive Regler den weiteren Vorteil, dass sie auf unerwartete Veränderungen des dynamischen Verhaltens der Regelstrecke, wie sie zum Beispiel bei Beschädigungen auftreten, reagieren und sich anpassen können. Lernfähige Reglerarchitekturen sind damit auch in der Lage, in beim Entwurf unvorhergesehenen Extremsituationen aktiv zur Sicherheit des geregelten Systems beizutragen.

Diese Sichtweise auf die Herausforderungen erhöhter Automatisierung bis hin zur Systemautonomie ermöglicht eine Einordnung der Problemstellung in das allgemeine Feld der Kybernetik¹. Diese wissenschaftliche Disziplin beschäftigt sich mit der fächerübergreifenden Betrachtung

¹Durch N. Wiener geprägter Ausdruck von der Lehre der Regelungs- und Steuerungsmechanismen, entlehnt aus dem engl. *cybernetics*, dieses aus dem gr. *kybernētikē*, Kunst des Steuerns, Steuermannskunst. Nach: Kluge - Etymologisches Wörterbuch der deutschen Sprache, bearb. von E. Seebold, Berlin ²³1999, S. 496.

der Steuerung und Regelung beliebiger dynamischer Systeme. Neben der klassischen Regelungstechnik können unter anderem auch die Robotik und die Künstliche Intelligenz in das breite Spektrum der Kybernetik eingeordnet werden. Die Regelungstechnik ermöglicht dabei geprägt durch mathematisch-physikalische Modellbetrachtungen die dezidierte Beeinflussung der dynamischen Größen eines Systems. Die Robotik und die Künstliche Intelligenz haben daneben das allgemeinformulierte Ziel, autonome Maschinen, respektive intelligente Agenten zu erschaffen. Hierbei repräsentiert die Robotik vor allem eine Gesamtsicht auf ein System bestehend aus Sensoren, Aktuatoren, Informationsverarbeitung und Struktur, während sich die Künstliche Intelligenz vor allem mit maschinellern Lernen und der Wissensrepräsentation befasst.

Betrachtet man nun ein unbemanntes Fluggerät, das über mehr als nur vollautomatisches Systemverhalten verfügt, so kombiniert dieses Elemente der Regelungstechnik, der Robotik und der Künstlichen Intelligenz unter den physikalischen Randbedingungen der luftfahrttechnischen Anwendung. Die eigentliche Regelung eines Flugzeuges stellt dabei ein verkoppeltes Mehrgrößenproblem dar, das mit existierenden linearen Ansätzen an jeweils gültigen Betriebspunkten unter dem Nachweis der Stabilität gelöst werden kann. Soll ein Regelkreis nun aber im Sinne der Künstlichen Intelligenz eine gewisse Lernfähigkeit aufweisen, müssen trotz des aufwändigeren Stabilitätsnachweises nichtlineare Regelungsansätze herangezogen werden. Zur Umsetzung von Lerneigenschaften in einem solchen Regelkreis bieten sich vor allem künstliche neuronale Netzwerke an, da sie informationstechnisch vergleichsweise einfach umzusetzen sind sowie nichtlineare Zusammenhänge schnell und genau erlernen und abbilden können. Hierbei besteht aus regelungstechnischer Sicht eine große Herausforderung darin, den Lernprozess der Netzwerke so zu gestalten, dass ein destabilisierender Einfluss auf den Regelkreis ausgeschlossen werden kann. Die vorliegende Arbeit leistet einen Beitrag zu genau dieser Problemstellung.

1.1 Stand der Technik

Künstliche neuronale Netzwerke (KNN) besitzen auf Grund ihrer Lernfähigkeit und ihres Aufbaus die Eigenschaft der Mustererkennung und der Abbildung nichtlinearer Zusammenhänge, was unter anderem in [1, 2] und [3] grundlegend dargestellt ist. Diese nachgewiesenen Approximationseigenschaften [4–6] werden häufig dann eingesetzt, wenn eine analytische Beschreibung von meist nichtlinearen Eigenschaften realer Systeme zu aufwändig oder wegen fehlender Kenntnis der Systemparameter gar unmöglich ist. Dies führt dazu, dass neuronale Netze in vielen wissenschaftlichen Bereichen angewendet werden, wie zum Beispiel der Prognose wirtschaftlicher Entwicklungen [7, 8], der geologischen Modellierung [9], der Beschreibung astronomischer Effekte [10], der Nutzung als virtuelle Sensorik [11], der Stützung integrierter Navigationssysteme [12, 13] und nicht zuletzt der nichtlinearen Regelung [14–17].

Da man das Lernen in einem KNN als mathematischen Optimierungsprozess auffassen kann, existiert eine Fülle von Quellen, die hierfür unterschiedliche Ansätze vorschlagen. Neben

dem häufig als Referenz genutzten gradientenbasierten Rückpropagationsalgorithmus [1, 2] sind auch Algorithmen mit höherer Ableitungsordnung verbreitet [18]. Außerdem findet man Ansätze, die bionisch inspirierte Optimierungsverfahren verwenden wie die Partikelschwarmoptimierung [19] oder evolutionäre Algorithmen [20, 21]. Letztere können dabei auch für die Optimierung der Netztopologie verwendet werden, was für Robustheit des Netzwerkverhaltens von Bedeutung ist und in einer Reihe von Untersuchungen zur optimalen Netzwerkarchitektur thematisiert wird [22, 23]. Neben Trainingsverfahren, die explizit die Bewahrung erlernten Wissens zum Ziel haben [24] existieren einige Arbeiten, bei denen die Lernmethoden regelungstechnisch motiviert sind [25, 26] oder aus der Theorie des Kalman-Filters abgeleitet werden [27]. Dabei ist zu beachten, dass wegen des zuweilen hohen Rechenaufwandes nicht alle Ansätze in gleicher Weise für eine Echtzeitanwendung in einem Regelkreis geeignet sind.

In der Luft- und Raumfahrttechnik sind besonders in der Flugregelung lineare Verfahren üblich, respektive aus Zulassungsgründen derzeit in der Anwendung gefordert [28]. Gleichwohl sind gerade bei der Bestimmung der aerodynamischen Beiwerte viele Effekte stark nichtlinear, wobei die aerodynamischen Eigenschaften direkt auf die Flugmechanik und -regelung eines Luftfahrzeuges wirken. Daher werden neuronale Netze auch zur Beiwertbestimmung [29, 30] oder zur Erweiterung und Optimierung linearer Regelungssysteme verwendet [31–33]. Umfangreiche Untersuchungen zu intelligenten Flugregelungssystemen in den USA, besonders während der 1990er und der 2000er Jahre, legen nahe, dass wirklich autonome Luft- und Raumfahrtsysteme lernfähiger Regelungsverfahren bedürfen und dass Zertifizierungs- und Zulassungskriterien für diese geschaffen werden müssen [17, 34–36]. Maßgebliche Vorteile dieser in der Regel KNN-basierten Regelungssysteme bestehen demnach im robusten Betrieb unter unbekannten Umgebungsbedingungen und einem hohen Sicherheitsgewinn durch die stabile Regelung auch von degradierten oder beschädigten Systemen.

Als Grundlage eines solchen nichtlinearen adaptiven Flugregelungssystems wird häufig das Prinzip der dynamischen Inversion, auch Eingangs-/Ausgangslinearisierung genannt, verwendet [37, 38]. Dieses modellbasierte Verfahren ermöglicht eine stabile Invertierung der Systemdynamik und damit ein dynamisches Regelungsverhalten bei weit gefassten Betriebsgrenzen. Um dabei auftretende Modellunsicherheiten auszugleichen, kann die dynamische Inversion mit neuronalen Netzwerken erweitert werden [39]. Diese Kombination aus modellbasierter Regelung und lernfähigem Element ist bereits in einer Vielzahl von Anwendungen für Luft- und Raumfahrzeuge nicht zuletzt auch im Flugversuch erfolgreich erprobt worden [40–49]. Dies gilt auch für Analysen hinsichtlich der Rekonfigurationseigenschaften bei Aktuatorausfällen [50] und der Regelung stark beschädigter Flugzeuge [51]. Die durch die Lernfähigkeit bedingte Toleranz gegenüber Modellunsicherheiten und Systemschäden kombiniert mit der Arbeitspunktunabhängigkeit des Regelungssystems hat in der Folge auch zu einer vermehrten Anwendung auf unbemannte Flugsysteme geführt [52–55]. Diese Nutzung bietet sich in besonderer Weise auch deswegen an, da kleinere unbemannte Fluggeräte als ideale Demonstrationsplattform für die Validierung dieses Regelungsansatzes im Flugversuch fungieren können. Dadurch

ist man in der Lage, weitere wichtige Erkenntnisse bezüglich der Zertifizierbarkeit nichtlinearer adaptiver Flugregler zu gewinnen, und es bietet sich die Möglichkeit, die allgemeine technische Akzeptanz von lernfähigen Regelungssystemen zu erhöhen.

1.2 Gegenstand der Arbeit

Die vorliegende Arbeit liefert einen Beitrag zur Nutzung künstlicher neuronaler Netzwerke in nichtlinearen adaptiven Flugregelungssystemen. Zu diesem Zweck werden zwei in ihrer Herangehensweise sehr unterschiedliche Regelungsansätze vorgestellt und analysiert: Die rein neuronale Regelung, die ohne ein physikalisches Modell der Regelstrecke auskommt, und die modellbasierte dynamische Inversion, die mit neuronalen Netzen erweitert wird. Diese Regelungsstrategien werden, wann immer möglich, aus einer allgemeinen Perspektive betrachtet, so dass eine Übertragung auch auf andere Regelungsprobleme möglich ist. Dabei wird auch ein statistisch motivierter Entwurfsprozess neuronaler Netzwerke miteinbezogen. Die Validierung der vorgestellten Verfahren soll am Beispiel zweier unbemannter Flugsysteme unterschiedlicher Größe erfolgen, welche in Abbildung 1.1 dargestellt sind: dem *Carolo P200* mit einer Startmasse von 5 kg sowie einer Spannweite von 200 cm und dem deutlich größeren *Carolo P360* mit einer Startmasse von 25 kg sowie einer Spannweite von 360 cm (siehe auch Tabelle 7.1. Die Anwendung der diskutierten Methoden, besonders der dynamischen Inversion, ist dabei jedoch nicht auf unbemannte System begrenzt, sondern kann auch auf bemannte Fluggeräte übertragen werden.



Abbildung 1.1: Die unbemannten Flugsysteme *Carolo P200* und *Carolo P360*.

Wichtiger Bestandteil dieser Arbeit ist ein Paradigmenwechsel hinsichtlich der Lernverfahren von zur Regelung genutzten neuronalen Netzwerken. Diese werden nicht länger als zu optimierendes Subsystem eines Regelkreises, sondern selbst als zu regelndes System betrachtet. Mit dieser Sichtweise ist es möglich, die Methode der Gleitzustandsregelung auf neuronale Lernverfahren anzuwenden. Dabei werden die Fehlerzustände eines neuronalen Netzes durch die Nutzung variabler Lernraten in den Gleitzustand gezwungen, der auf Grund seiner Eigenschaften zu global asymptotischer Stabilität des Netzwerkes führt. Daraus abgeleitet, besteht ein maßgeblicher Beitrag dieser Arbeit aus der Anwendung des Gleitzustandslernverfahrens

auf die dynamische Inversion. Wegen ihres Aufbaus werden nämlich nicht nur die Fehlerzustände der verwendeten Netzwerke in den Gleitzustand gebracht, sondern auch die Zustände der Fehlerdynamik des neuronal beeinflussten geschlossenen Regelkreises. Durch die intrinsische Stabilitätsforderung des Lernverfahrens mit Gleitzustand ist eine neue Perspektive auf die Ljapunow-basierte Stabilitätsanalyse dieses Regelungsverfahrens möglich. Gleichzeitig wird gezeigt, dass das Regelungssystem auch die positive Eigenschaft des Gleitzustandes von hoher Robustheit gegenüber äußeren Störungen aufweist. Vergleichende Analysen im Rahmen nichtlinearer Simulationen zeigen weiterhin, dass der gewählte Ansatz dem üblicherweise verwendeten Lernverfahren des Rückpropagationsalgorithmus' in seiner Leistungsfähigkeit überlegen ist. Dies wird auch an Untersuchungen von Flugzeugen mit Beschädigungen deutlich, bei denen das vorgestellte Lernverfahren eine höhere Toleranz gegenüber Systemschäden aufweist. Zusammengefasst leistet die vorliegende Arbeit damit einen Beitrag zur regelungstechnischen Transparenz künstlicher neuronaler Netzwerke und ermöglicht damit eine verbesserte Betrachtung lernfähiger Regelungssysteme.

1.3 Gliederung der Ausarbeitung

Da der Schwerpunkt dieser Arbeit auf der regelungstechnischen Anwendung neuronaler Netzwerke liegt, werden in Kapitel 2 zunächst alle wichtigen Grundlagen vorwärtsgerichteter Netze und ihrer Lernverfahren dargestellt. Dabei wird insbesondere ein systematischer Ansatz des Netzwerktrainings vorgestellt, der auf Grund seiner statistischen Methodik eine gezielte Bewertung des Lernerfolges ermöglicht. Das anschließende Kapitel beschäftigt sich mit der Anwendung der Prinzipien der strukturvariablen Regelung und des Gleitzustandes auf Lernverfahren neuronaler Netze. Es wird gezeigt, dass sich durch die Betrachtung neuronaler Netze als geregelte Systeme besondere Stabilitätsforderungen ableiten lassen, die erhebliche Vorteile bei der regelungstechnischen Anwendung zur Folge haben.

Nachdem in Kapitel 4 kurz auf die Aspekte der Flugsimulation und Reglerimplementierung eingegangen wird, widmet sich Kapitel 5 dem Konzept der rein neuronalen adaptiven Flugregelung. Dieser Regelungsansatz bietet den Vorteil, auf ein physikalisches Modell des zu regelnden Systems verzichten zu können. Im Gegensatz dazu wird im darauffolgenden Kapitel die stark modellbasierte dynamische Inversion als nichtlineares Regelungsverfahren vorgestellt. Diese Regelungsstrategie wird im Sinne der Adaptivität mit neuronalen Netzen erweitert, wobei ein besonderer Schwerpunkt auf der Anwendung des in Kapitel 3 vorgestellten Lernverfahrens liegt. In Kapitel 7 werden die vorgestellten Verfahren im Rahmen nichtlinearer Simulationen auf ihre Leistungsfähigkeit unter Störeinfluss untersucht. Dabei kommen zwei unterschiedliche unbemannte Flugsysteme zum Einsatz, die auch unter der Randbedingung von Systemschäden analysiert werden. Das letzte Kapitel fasst die zentralen Ergebnisse dieser Arbeit zusammen und bewertet deren regelungstechnische Relevanz. Die Ausarbeitung schließt mit einem Ausblick auf mögliche Verbesserungen und weitere Forschungsansätze zur Nutzung neuronaler Netzwerke in lernfähigen Flugregelungssystemen.

2 Grundlagen neuronaler Netzwerke

Künstliche neuronale Netzwerke gelten als ein grundlegender Bestandteil des Forschungsfeldes, das landläufig als Künstliche Intelligenz (KI) bezeichnet wird. Sie stellen den Versuch dar, bestimmte Eigenschaften biologischer Nervenzellen, besonders die der Lernfähigkeit, auf technische Systeme zu übertragen. Eine der ersten Arbeiten, die in diesem Zusammenhang der KI zugeordnet wird, ist die Darstellung eines künstlichen Neuronenmodells von Warren McCulloch und Walter Pitts aus dem Jahre 1943 [56]. Weitere wichtige Entwicklungsschritte sind unter anderem die erste Lernregel von Donald Hebb [57] sowie die Entwicklung des sogenannten Perzeptrons von Frank Rosenblatt [58]. Auch wenn diese ersten Ansätze in ihren Eigenschaften deutliche Grenzen aufweisen, stellen sie dennoch zusammengekommen die Grundlagen der Theorie der künstlichen neuronalen Netze dar.

Das Ziel der Künstlichen Intelligenz lässt sich nach [59] allgemein beschreiben als der Versuch, Maschinen zu erstellen, die in komplexen, sich ändernden Systemen selbstständig arbeiten. Auch wenn neuronale Netze derzeit nicht in der Lage sind, diese sehr allgemeine Zielsetzung für jedes technische System in beliebiger Umwelt zu realisieren, so stellt ihre Lernfähigkeit doch eine besondere Eigenschaft dar, zumindest Teile dessen zu adressieren. Denn die aus der Lernfähigkeit ableitbaren Eigenschaften der Mustererkennung und der Approximation nichtlinearer Zusammenhänge ermöglichen letztlich auch das Erfassen regelungstechnischer Muster und Prozesse. Im Hinblick auf die Umsetzung adaptiver, fehlertoleranter Flugregelungsverfahren hilft eine genaue Analyse dieser Eigenschaften künstlicher neuronaler Netze, um ihre Einsatzmöglichkeiten auf diesem Gebiet zu bewerten.

2.1 Biologisches Vorbild und technisches Neuronenmodell

Um vom biologischen Vorbild zu einer technischen Umsetzung zu gelangen, ist es erforderlich, eine Modellbeschreibung zu finden, die die Wirklichkeit in angemessener Weise abbildet. In diesem Zusammenhang lässt sich feststellen, dass von neurophysiologischer Seite die Beschreibung neuronaler Prozesse seit den Anfängen der KI erheblich genauer geworden ist. Demnach basiert die Informationsverarbeitung von Neuronen nicht nur auf ihren elektrophysiologischen Eigenschaften, also der synaptischen Aktivität, sondern auch in starkem Maße auf den intrinsischen Parametern, also ihren dynamischen Eigenschaften [60]. Diese nichtlinearen Systemeigenschaften der Neuronen befinden sich nahe der sogenannten Bifurkation; das heißt eine kleine Änderung eines Parameters, der das dynamische Verhalten bestimmt, hat eine

plötzliche qualitative Änderung der Zustandsgrößen des Systems zur Folge. Daraus folgt, dass zwei Neuronen in der gleichen Hirnregion auf dieselbe synaptische Eingabe unterschiedlich reagieren können. Aktuelle Neuronenmodelle beschreiben dieses Systemverhalten durch nicht-lineare Differentialgleichungen vierter Ordnung [60], was die Komplexität eines verschalteten Netzwerkes als Gesamtsystem in hohem Maße steigert.

Biologische neuronale Netzwerke stellen einen Schaltkreis vieler parallel verknüpfter Elemente, den Nervenzellen, dar, wobei die Größe des Neuronenverbandes die Komplexität des resultierenden dynamischen Systems bestimmt. Der Signalfluss zwischen einzelnen Neuronen verläuft dabei über gewichtete Verbindungen. Jede dieser Nervenzellen besteht aus einem Zellkörper, dem Soma, das den Zellkern enthält, und aus dem sich die Eingänge, die sogenannten Dendriten, verzweigen. Dieser Dendritenbaum kann Eingangssignale von mehreren tausend Neuronen empfangen. Daneben besteht ein Neuron aus einer langen Nervenfasern, auch Axon genannt. Am Ende des Axons befinden sich bis zu 10^3 Synapsen, die die Signalübertragung zu den Dendriten nachgeschalteter Neuronen übernehmen, wodurch eine massiv parallele Verarbeitung von Informationen entsteht. Die Signal- bzw. Informationsübertragung selbst erfolgt in Form verschiedener elektrochemischer Reaktionen, die in [1] detailliert beschrieben sind. Über die Synapsen werden gewichtete Eingangssignale an die Dendriten geleitet, die im Soma zu einem Aktionspotential aufsummiert werden, wodurch der Aktivierungszustand eines Neurons bestimmt ist. Wird durch dieses Aktionspotential ein bestimmter Schwellenwert überschritten, entsteht im Soma ein Impuls, der über das Axon und seine Synapsen an die Dendriten nachgeschalteter Neuronen geleitet wird. Die Synapsen haben ihrerseits entweder einen verstärkenden oder einen hemmenden Einfluss auf die Signalstärke, so dass ihre Wirkung auf die Aktionspotentiale der verschalteten Neuronen und damit auf deren Aktivität das Systemverhalten maßgeblich definiert. Ein Lernprozess, der also dieses Systemverhalten beeinflusst, verändert die verstärkenden oder hemmenden Eigenschaften der Synapsen.

Im diesem Kontext stellt sich die Frage, wie detailliert das Modell der realen Mechanismen beschaffen sein soll, damit die beschriebenen Eigenschaften im bionischen Sinne in einem vereinfachten technischen Modell umgesetzt werden können. Die aktuellen Neuronenmodelle gehen dabei in ihrem Detailgrad deutlich über die für KNN gängigen in [1, 2] und [59] beschriebenen Abläufe hinaus. Letztere beschreiben aber gerade auf abstrakter Ebene die Dynamik neuronaler Prozesse sehr anschaulich und begünstigen so die regelungstechnische Umsetzung. Daher soll diese vereinfachte Betrachtungsweise übernommen und im Folgenden skizziert werden.

Künstliche neuronale Netze bestehen aus idealisierten Neuronenmodellen, die in Schichten angeordnet sind, wobei diese mit dem Index (l) gekennzeichnet sind. Jedes i -te von $q^{(l)}$ Neuronen einer Schicht (l) verfügt über eine Transferfunktion $f_i^{(l)}$, die dessen aktuellen Aktivierungszustand $n_i^{(l)}$, der von den gewichteten Eingaben abhängt, in eine Ausgabe $o_i^{(l)}$ überführt.

$$o_i^{(l)} = f_i^{(l)}(n_i^{(l)}) \quad (2.1)$$

In Abbildung 2.1 ist ein in einem Netzwerk verschaltetes Neuron dargestellt, das seine Eingaben von Neuronen der vorangehenden Schicht $(l-1)$ erhält und diese weiterpropagiert.

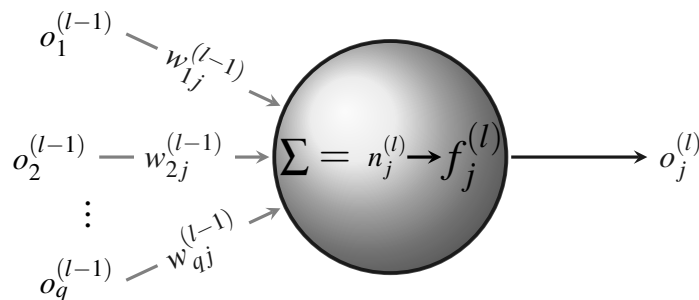


Abbildung 2.1: Darstellung eines künstlichen Neurons.

Die Verbindungsgewichte $w_{ij}^{(l)}$ werden der Schicht zugeordnet, von der sie ausgehen, i bezeichnet das Start- und j das Zielneuron. Die Vorwärtspropagation des Neurons aus Abbildung 2.1 lässt sich demnach folgendermaßen schreiben:

$$o_j^{(l)} = f_j^{(l)}(n_j^{(l)}) = f_j^{(l)}\left(\sum_{i=1}^{q^{(l-1)}} o_i^{(l-1)} \cdot w_{ij}^{(l-1)}\right). \quad (2.2)$$

2.1.1 Transferfunktionen

Die Transferfunktionen der einzelnen Neuronen haben starken Einfluss auf deren Signalverarbeitung und damit auf die Gesamtausgabe eines Netzwerkes, da diese den Aktivierungszustand in eine Ausgabe überführen. Sie definieren folglich, ob das Neuron auf Basis der eingehenden Reize seinerseits feuert und wenn ja, in welcher Form. Es gibt eine Fülle verschiedener Transferfunktionen [1, 2, 7], wobei nicht alle für die regelungstechnische Anwendung in gleichem Maße geeignet sind. In Abbildung 2.2 sind die lineare Identitätsfunktion (a) und eine Sigmoidfunktion (b) dargestellt, da diese bestimmte Anforderungen erfüllen und im weiteren Verlauf verwendet werden sollen.

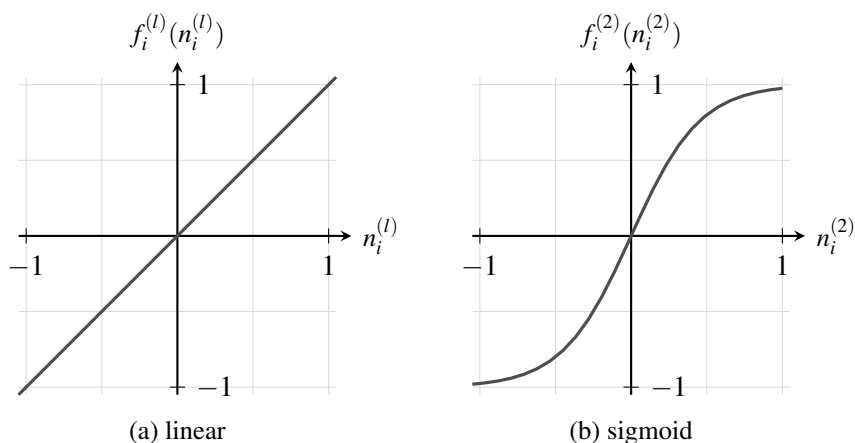


Abbildung 2.2: Typische Transferfunktionen neuronaler Netze.

Eine wichtige Eigenschaft für die Echtzeitanwendung ist die stetige Differenzierbarkeit, da die meisten dafür verwendeten Lernmethoden gradientenbasierte Optimierungsverfahren darstellen, was in Abschnitt 2.3 deutlich wird. Darüber hinaus sollen KNN nichtlineare Zusammenhänge approximieren, so dass zumindest in einer verdeckten Schicht (bei einem dreischichtigen Netz mit dem Index 2 gekennzeichnet) eine nichtlineare Transferfunktion verwendet werden muss. Die Identitätsfunktion überführt, wie der Name sagt, eine Eingabe in dieselbe Ausgabe; darüber hinaus erleichtert sie die Stabilitätsbetrachtung der neuronalen Regelungselemente, was in Kapitel 6 von Vorteil sein wird. Sigmoidale Funktionen besitzen eine S-förmige Gestalt und weisen ab einem bestimmten Aktivierungszustand asymptotische Sättigungsbereiche auf. Im weiteren Verlauf wird als sigmoide Funktion ausschließlich die Tangens Hyperbolicus-Funktion verwendet; diese ist in Gleichung 2.3, ihre Ableitung in Gleichung 2.4 definiert.

$$f(n) = \frac{2}{1 + e^{-2n}} - 1 \quad (2.3)$$

$$f'(n) = \frac{4e^{-2n}}{(1 + e^{-2n})^2} \quad (2.4)$$

2.2 Topologien neuronaler Netzwerke

Die Topologie eines künstlichen neuronalen Netzes besteht aus einzelnen Schichten, in denen eine beliebige Anzahl von Neuronen angeordnet sein kann. Neben der Ein- und Ausgabeschicht verfügt ein KNN üblicherweise über eine, zuweilen auch mehrere verdeckte Schichten. Die Signalverarbeitung in der grundlegendsten Netzwerkarchitektur verläuft dabei nur vorwärtsgerichtet (engl. *feed forward*) von der Ein- zur Ausgabeschicht, wobei die Neuronen einer Schicht mit allen Neuronen der dahinterliegenden verbunden sind. Darüber hinaus existieren Varianten vorwärtsgerichteter Netze, in denen schichtübergreifende Verbindungen (engl. *shortcuts*) eingefügt werden. Daneben gibt es Topologien mit verschiedenen Arten der Rückkopplung (engl. *recurrent networks*): die direkte Rückkopplung des Neuronenausgangs auf den Eingang, die laterale Rückkopplung in einer Schicht, sowie die indirekte Rückkopplung zwischen verschiedenen Schichten. Detaillierte Übersichten dieser Architekturen sind in [2] und [7] zu finden. Um eine innere zeitabhängige Dynamik im neuronalen Modell zu erzeugen, werden häufig auch zeitverzögerte Netzwerke (engl. *time delay networks*) verwendet, bei denen die direkte Rückkopplung des Ausgangs auf den Eingang mit einer zusätzlichen Zeitverzögerung behaftet ist. All diese Varianten haben das Ziel, durch die Einführung zusätzlicher Freiheitsgrade verbesserte Abbildungseigenschaften der Netzwerke zu erzeugen, was jedoch auch ihre Komplexität erhöht. Die weiteren Betrachtungen in dieser Arbeit konzentrieren sich auf vorwärtsgerichtete Netzwerke mit einer verdeckten Schicht (in wenigen Ausnahmen zwei verdeckte Schichten) mit sigmoidaler Transferfunktion, einem Neuron in der Ausgabeschicht sowie linearen Transferfunktionen in der Ein- und Ausgabeschicht. Ein solches Netzwerk ist in Abbildung 2.3 dargestellt, wobei die Eingabe durch den Vektor \vec{x} und die Ausgabe durch y dargestellt sind.

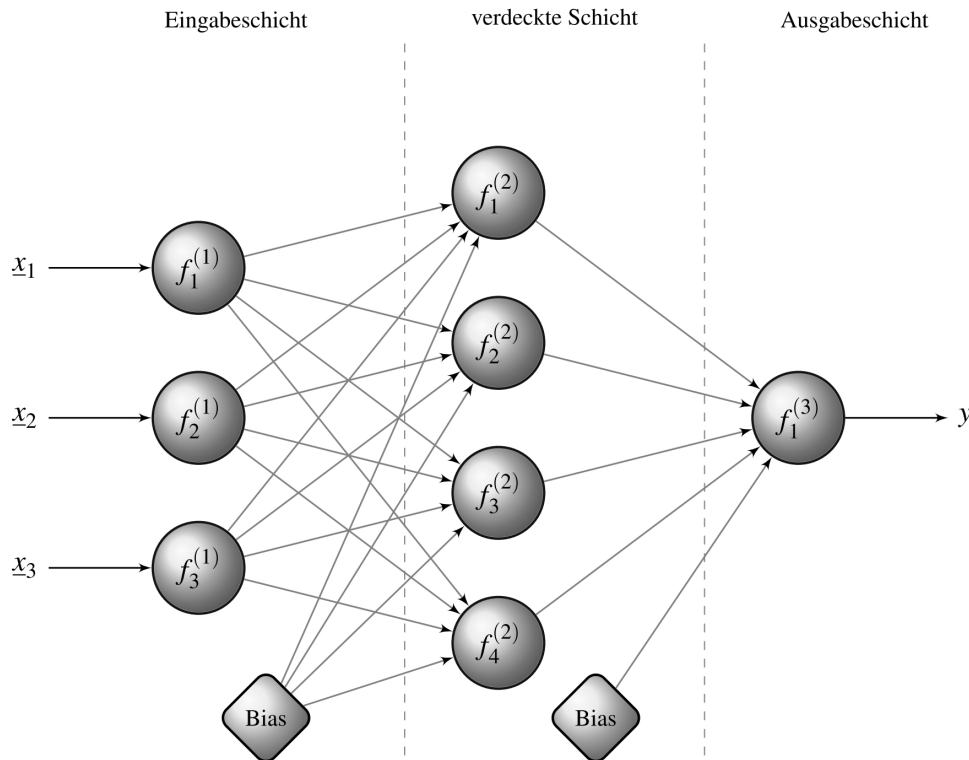


Abbildung 2.3: Darstellung eines dreischichtigen Netzes mit einem Ausgabeneuron.

Dieses Netzwerk verfügt über zusätzliche Bias-Werte, Elemente mit einer konstanten Ausgabe von 1, die gewichtet in das Netzwerk gegeben werden, wodurch weitere Freiheitsgrade zur Netzanpassung gegeben sind. Die Gewichte der Bias-Werte werden mit b indiziert. In Anlehnung an die in Abschnitt 2.1 eingeführte Indizierung der Netzwerkgewichte lassen sich die Gewichtsmatrizen eines dreischichtigen Netzes mit Gleichung 2.5 darstellen. Dabei ist zu beachten, dass diese Gleichung den allgemeinen Fall darstellt, bei einem Netzwerk mit nur einer Ausgabe reduziert sich die Matrix $\mathbf{w}^{(2)}$ auf den ersten Spaltenvektor.

$$\mathbf{w}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \cdots & w_{1q^{(2)}}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q^{(1)}1}^{(1)} & w_{q^{(1)}2}^{(1)} & \cdots & w_{q^{(1)}q^{(2)}}^{(1)} \\ w_{b1}^{(1)} & w_{b2}^{(1)} & \cdots & w_{bq^{(2)}}^{(1)} \end{bmatrix}^T, \quad \mathbf{w}^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & \cdots & w_{1q^{(3)}}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q^{(2)}1}^{(2)} & w_{q^{(2)}2}^{(2)} & \cdots & w_{q^{(2)}q^{(3)}}^{(2)} \\ w_{b1}^{(2)} & w_{b2}^{(2)} & \cdots & w_{bq^{(3)}}^{(2)} \end{bmatrix}^T \quad (2.5)$$

Diese beiden Gewichtsmatrizen lassen sich nun in einer allgemeinen Gewichtsmatrix \mathbf{w} zusammenfassen, was im weiteren Verlauf zur Verdeutlichung der Schreibweise genutzt werden wird.

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}^{(2)} & 0 \\ 0 & \mathbf{w}^{(1)} \end{bmatrix} \quad (2.6)$$

Fasst man nun die Gleichungen 2.1 und 2.5 in allgemeiner Vektorschreibweise zusammen, er-

gibt sich für die Gesamtausgabe eines vorwärtsgerichteten Netzwerkes folgender Ausdruck:

$$\vec{y} = \vec{f}^{(3)} \left(\mathbf{w}^{(2)} \cdot \vec{f}^{(2)} \left[\mathbf{w}^{(1)} \cdot \vec{f}^{(1)}(\vec{n}^{(1)}) \right] \right). \quad (2.7)$$

Hierzu sei angemerkt, dass der Vektor \vec{x} und die jeweiligen Vektoren der Schichteingänge bei der Nutzung von Bias-Werten um eine 1 verlängert werden müssen. Da außerdem gemäß Abbildung 2.3 lineare Transferfunktionen für $\vec{f}^{(1)}$ und $\vec{f}^{(3)}$ und nur ein Ausgabeneuron verwendet werden, folgen einige Vereinfachungen: Die Aktivität der Neuronen der Eingabeschicht $\vec{n}^{(1)}$ ist gleich dem Eingabevektor \vec{x} und der Ausgabevektor \vec{y} reduziert sich zu einem Skalar. Dies liefert die Gesamtausgabe des Netzes mit:

$$y = \mathbf{w}^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}^{(1)} \cdot \vec{x}). \quad (2.8)$$

2.3 Lernverfahren

Die bereits skizzierten Effekte in biologischen neuronalen Netzen besagen, dass ein Lernprozess eine Veränderung der synaptischen Verbindungen zur Folge hat. Übertragen auf ein KNN muss ein Lernalgorithmus die Gewichtsmatrizen verändern, und zwar dergestalt, dass eine gewünschte Zielfunktion abgebildet wird; das Netz wird gewissermaßen trainiert. In allgemeiner Form kann man einen Lernprozess mit Gleichung 2.9 beschreiben.

$$w_{ij,neu}^{(l)} = w_{ij,alt}^{(l)} + \Delta w_{ij}^{(l)} \quad (2.9)$$

Generell lassen sich neuronale Lernverfahren in überwachte und unüberwachte Ansätze einteilen, wobei die überwachten sich zusätzlich in bestärkende und korrigierende Verfahren unterscheiden lassen [1, 2, 7]. Das wohl bekannteste unüberwachte Lernverfahren ist die Hebb'sche Lernregel, welche besagt, dass zwei Neuronen, die gleichzeitig aktiv sind, eine höhere Interaktion aufweisen sollen als solche, deren Aktivität unkorreliert ist [1]. Die Änderung eines Verbindungsgewichtes basiert demnach auf den Ausgaben beider verbundener Neuronen. Bei überwachten Lernverfahren wird generell eine Aussage über die Qualität der Netzwerkausgabe gemacht, wobei bestärkende Verfahren lediglich eine Aussage darüber treffen, ob die Ausgabe gut oder schlecht gewesen ist. Bei korrigierenden Verfahren wird der Ausgabefehler genau bestimmt und durch das Netzwerktraining minimiert. Mathematisch gesehen entspricht dies einem hochdimensionalen Optimierungsproblem. Da für die regelungstechnische Anwendung eine möglichst schnelle Minimierung einer quantifizierbaren Zielfunktion, nämlich des Regelfehlers, erforderlich ist, bieten sich korrigierende Verfahren hierfür folglich an. Das bekannteste dieser Verfahren ist der gradientenbasierte Rückpropagationsalgorithmus (engl. *backpropagation algorithm*), von dem eine Vielzahl verschiedener Varianten vorgeschlagen wurde [1, 2, 25, 61]. Daneben sind auch bionische Verfahren zur Einstellung der Verbindungsgewichte denkbar, etwa die Partikelschwarmoptimierung [19] oder evolutionäre Algorithmen [20]. Auf Grund des relativ hohen Rechenaufwandes sind diese Verfahren für die regelungstechnische Verwendung von KNN allerdings meist ungeeignet. Der Ansatz, Prinzipien der nichtlinearen strukturvariablen Regelung für das Training neuronaler Netze zu nutzen, wird in Kapitel 3 im Detail erläutert.

2.3.1 Lernen durch Rückpropagation des Fehlers

Das Ziel des Rückpropagationsalgorithmus' ist es, die Netzwerkgewichte in Verbindung mit einer Zielfunktion optimal einzustellen. Dazu wird eine zu minimierende Funktion definiert, die in Abhängigkeit der Verbindungsgewichte steht. Hierfür wird meist der quadratische Fehler zwischen kommandierter Ausgabe \vec{y}_k und aktueller Ausgabe \vec{y} gewählt:

$$E = \frac{1}{2} (\vec{y}_k - \vec{y})^T \cdot (\vec{y}_k - \vec{y}) . \quad (2.10)$$

Die Abweichung zwischen Soll- und Ist-Ausgabe stellt den Netzfehler $\vec{\epsilon}$ dar.

$$\vec{\epsilon} = \vec{y}_k - \vec{y} \quad (2.11)$$

Um die Verbindungsgewichte anzupassen, ist es nötig, deren Einfluss auf den Ausgabefehler zu quantifizieren. Dazu werden die partiellen Ableitungen der Fehlerfunktion E nach allen Gewichten $w_{ij}^{(l)}$ gebildet und diese mit Hilfe der Kettenregel umformuliert.

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \frac{\partial E}{\partial n_j^{(l+1)}} \cdot \frac{\partial n_j^{(l+1)}}{\partial w_{ij}^{(l)}} \quad (2.12)$$

Mit Gleichung 2.2 lässt sich dieser Ausdruck umschreiben zu:

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \frac{\partial E}{\partial n_j^{(l+1)}} \cdot \frac{\partial}{\partial w_{ij}^{(l)}} \sum_{i=0}^{q^{(l)}} o_i^{(l)} \cdot w_{ij}^{(l)} = \frac{\partial E}{\partial n_j^{(l+1)}} \cdot o_i^{(l)} . \quad (2.13)$$

Zusätzlich wird der rückpropagierte Fehler $\delta_j^{(l)}$ eingeführt, der den Einfluss der Aktivität jedes Neurons einer Schicht auf die Fehlerfunktion beschreibt.

$$\delta_j^{(l)} = \frac{\partial E}{\partial n_j^{(l)}} \quad (2.14)$$

Dieser lässt sich durch erneute Anwendung der Kettenregel in allgemeiner Form darstellen:

$$\delta_j^{(l)} = \frac{\partial E}{\partial n_j^{(l)}} = \frac{\partial E}{\partial o_i^{(l)}} \cdot \frac{\partial o_i^{(l)}}{\partial n_j^{(l)}} = \frac{\partial E}{\partial o_i^{(l)}} \cdot \frac{\partial f_i^{(l)}(n_j^{(l)})}{\partial n_j^{(l)}} = \frac{\partial E}{\partial o_i^{(l)}} \cdot f_i^{\prime(l)} . \quad (2.15)$$

Für die Ausgabeschicht L lässt sich der rückpropagierte Fehler $\delta_j^{(L)}$ mit Hilfe der Netzausgabe y_i und der Ableitung der Identitätsfunktion schreiben als:

$$\delta_j^{(L)} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial n_j^{(L)}} = \frac{\partial E}{\partial y_i} \cdot f_i^{\prime(L)} = -(y_{k,i} - y_i) . \quad (2.16)$$

Zur Fortsetzung der Rückpropagation ist der rückpropagierte Fehler der Ausgabeschicht nun bekannt, so dass unter erneuter Verwendung der Kettenregel $\delta_j^{(l)}$ für jedes beliebige Neuron

bestimmt werden kann. Dies ist in Gleichung 2.17 dargestellt, so dass nun alle Informationen bekannt sind, um Gleichung 2.12 zu lösen.

$$\delta_j^{(l)} = \frac{\partial E}{\partial n_j^{(l)}} = \sum_{i=1}^{q^{(l+1)}} \left(\frac{\partial E}{\partial n_i^{(l+1)}} \cdot \frac{\partial n_i^{(l+1)}}{\partial o_j^{(l)}} \cdot \frac{\partial o_j^{(l)}}{\partial n_j^{(l)}} \right) = f_j'^{(l)} \cdot \sum_{i=1}^{q^{(l+1)}} \left(\delta_i^{(l+1)} \cdot w_{ji}^{(l)} \right) \quad (2.17)$$

Im Sinne einer übersichtlicheren Darstellung lässt sich der hergeleitete Zusammenhang mit Hilfe von Gleichung 2.13 für jede Schicht des KNN auch in Matrixschreibweise darstellen.

$$\frac{\partial E}{\partial \mathbf{w}^{(l)}} = \left[\vec{\delta}^{(l+1)} \cdot \vec{o}^{(l)T} \right]^T \quad (2.18)$$

Dabei kann in Anlehnung an Gleichung 2.17 der Vektor der rückpropagierten Fehler einer Schicht berechnet werden.

$$\vec{\delta}^{(l)} = \text{diag} \left(\vec{f}'^{(l)} \right) \cdot \mathbf{w}^{(l)} \cdot \vec{\delta}^{(l+1)} \quad (2.19)$$

Daneben existiert eine weitere Möglichkeit, den Einfluss der Verbindungsgewichte auf das Verhalten des Netzwerkes zu charakterisieren. Dabei betrachtet man die partiellen Ableitungen des Netzausganges nach den Verbindungsgewichten. Dieser Ansatz bietet Vorteile bei der praktischen Implementierung erweiterter Lernverfahren und zusätzlicher Netzwerkfunktionen wie dem gleitzustandsbasierten Training aus Kapitel 3 oder dem neuronalen Systembeobachter, der in Kapitel 5 erläutert wird. Aus Gleichung 2.20 ist ersichtlich, dass dieser Ansatz mit der Ableitung der Fehlerfunktion nach den Verbindungsgewichten mathematisch identisch ist.

$$\frac{\partial E}{\partial \mathbf{w}} = \frac{\partial E}{\partial \vec{y}} \cdot \frac{\partial \vec{y}}{\partial \mathbf{w}} = - \left[\left(\frac{\partial \vec{y}}{\partial \mathbf{w}} \right)^T \cdot (\vec{y}_k - \vec{y}) \right]^T = - \left[\left(\frac{\partial \vec{y}}{\partial \mathbf{w}} \right)^T \cdot \vec{\epsilon} \right]^T \quad (2.20)$$

Die in Gleichung 2.20 eingeführte Ableitung des Netzausganges nach den Gewichten lässt sich in Anlehnung an Gleichung 2.18 schreiben als:

$$\frac{\partial \vec{y}}{\partial \mathbf{w}^{(l)}} = \left[\vec{\delta}^{(l+1)} \cdot \vec{o}^{(l)T} \right]^T. \quad (2.21)$$

Dabei bezeichnet $\hat{\delta}_j^{(l)}$ die Empfindlichkeit des Ausganges gegenüber dem Eingang des Neurons j in Schicht l . Bezogen auf ein Ausgangssignal m lässt sich diese analog zu Gleichung 2.14 elementweise bestimmen mit

$$\hat{\delta}_{j,m}^{(l)} = \frac{\partial y_m}{\partial n_j^{(l)}}, \quad (2.22)$$

so dass man letztlich in Matrixschreibweise folgenden Zusammenhang erhält:

$$\vec{\hat{\delta}}^{(l)} = \text{diag} \left(\vec{f}'^{(l)} \right) \cdot \mathbf{w}^{(l)} \cdot \vec{\hat{\delta}}^{(l+1)}. \quad (2.23)$$

Für die Ausgangsschicht L lässt sich die Empfindlichkeit der jeweiligen Ausgangsneuronen m nach Gleichung 2.24 bestimmen, so dass alle Informationen zur Bestimmung der Ableitung des Ausganges nach den Verbindungsgewichten gegeben sind.

$$\hat{\delta}_{j,m}^{(L)} = f_m'^{(L)} \quad (2.24)$$

Nachdem nun der Einfluss der Verbindungsgewichte auf den Ausgabefehler des Netzwerkes und die daraus abgeleitete Fehlerfunktion quantifiziert ist, kann unter Einführung der Lernrate μ die Anpassung der Verbindungsgewichte in negativer Gradientenrichtung erfolgen:

$$\Delta \mathbf{w} = -\mu \cdot \frac{\partial E}{\partial \mathbf{w}} = -\mu \cdot \nabla E(\mathbf{w}) . \quad (2.25)$$

Hierbei enthält $\nabla E(\mathbf{w})$ die partiellen Ableitungen der Fehlerfunktion nach allen Verbindungsgewichten. Verknüpft man außerdem die zuvor getroffenen Aussagen mit Gleichung 2.5, so ist es möglich, die Änderung der Verbindungsgewichte eines dreischichtigen Netzes für die einzelnen Gewichtsmatrizen aufzuschlüsseln. Dies ist in den Gleichungen 2.26 und 2.27 dargestellt und wird in Kapitel 6 von Bedeutung sein.

$$\Delta \mathbf{w}^{(1)} = \mu^{(1)} \cdot \left[\vec{x} \cdot [\vec{y}_k - \vec{y}] \cdot \mathbf{w}^{(2)} \cdot \text{diag} \left(\vec{f}'^{(2)} \right) \right]^T \quad (2.26)$$

$$\Delta \mathbf{w}^{(2)} = \mu^{(2)} \cdot \left[\vec{f}^{(2)} \left(\mathbf{w}^{(1)} \cdot \vec{x} \right) \cdot [\vec{y}_k - \vec{y}]^T \right]^T \quad (2.27)$$

2.3.2 Gradientenabstieg mit zusätzlichem Moment

Der in Gleichung 2.25 beschriebene Rückpropagationsalgorithmus mit konstanter Lernrate μ erlaubt die Bewegung entlang des Gradienten der Fehlerfläche in Richtung eines Minimums. Die Schrittweite wird demnach durch die Wahl der Lernrate und durch den Betrag des Gradienten bestimmt. Ist die Schrittweite hoch, erfolgt eine schnelle Bewegung auf der Fehlerfläche, was allerdings zum Überspringen guter Minima (lokal oder global) oder zu Oszillation an steilen Tälern der Fehlerfläche führen kann; ist die Schrittweite klein, reduziert dies die Konvergenzgeschwindigkeit des Algorithmus'. Besonders auf Plateaus der Fehlerfläche, wo der Gradient gegen Null geht, führt eine zu niedrig gewählte konstante Lernrate nahezu zum Stillstand der Optimierung.

Eine einfache Möglichkeit, diesem Problem entgegenzuwirken, ist die Einführung eines sogenannten Momententerms. Dieses Moment erweitert die Gewichtsänderung zum Zeitpunkt t mit einem zusätzlichen, durch den Koeffizienten σ definierten, Anteil der Gewichtsänderung zum Zeitpunkt $t - 1$ [1, 7]:

$$\Delta \mathbf{w}_t = -\mu \cdot \nabla E(\mathbf{w})_t + \sigma \cdot \Delta \mathbf{w}_{t-1} . \quad (2.28)$$

Dieses Lernverfahren folgt also nicht der direkten negativen Gradientenrichtung, sondern bestimmt eine Art gewichtetes Mittel des aktuellen Gradienten und der vorangegangenen Gewichtsänderung [1]. Dadurch wird erreicht, dass die Schrittweite auf flachen Plateaus erhöht und an steilen Tälern reduziert wird. Der Nachteil dieses Ansatzes ist hingegen die Einführung des zusätzlichen Auslegungsparameters σ . Dieser wird in der Regel im Intervall $[0; 1]$ definiert, sollte aber gerade bei der Verwendung von in Echtzeit lernenden KNN genau wie die Lernrate μ im Sinne der Stabilität nicht zu hoch gewählt werden.

2.3.3 Der Levenberg-Marquardt-Algorithmus

Um die Lerngeschwindigkeit von KNN deutlich zu erhöhen, sind Verfahren höherer Ordnung wie das nach Levenberg-Marquardt entwickelt worden. Die Grundlage dieses Algorithmus', der eine höhere Konvergenzgeschwindigkeit als der in Abschnitt 2.3.1 beschriebene Ansatz aufweist, ist das sogenannte Gauß-Newton-Verfahren, welches eine Erweiterung des klassischen Newton-Verfahrens darstellt [18]. Das Newton-Verfahren selbst ist ein Verfahren zweiter Ordnung, das neben dem Gradienten auch Informationen über die Krümmung der Fehlerfunktion bei jedem Rechenschritt einbezieht. Dabei wird eine Taylor-Entwicklung der zu approximierenden Funktion durchgeführt und nach dem zweiten Glied abgebrochen. Mit dem Ansatz aus [62] kann man dies auf die Fehlerfunktion eines neuronalen Netzes übertragen.

$$E(\mathbf{w} + \Delta\mathbf{w}) = E(\mathbf{w}) + \nabla E(\mathbf{w}) \cdot \Delta\mathbf{w} + \frac{1}{2} \Delta\mathbf{w}^T \cdot \nabla^2 E(\mathbf{w}) \cdot \Delta\mathbf{w} \quad (2.29)$$

Hierbei wird $\nabla^2 E(\mathbf{w})$ auch als Hesse-Matrix $\mathbf{H}(\mathbf{w})$ bezeichnet, welche die zweiten partiellen Ableitungen der Fehlerfunktion nach den Verbindungsgewichten enthält. Aus diesem Zusammenhang lässt sich nach [1] der Gradient der obigen Funktion bestimmen zu:

$$\nabla E(\mathbf{w} + \Delta\mathbf{w})^T \approx \nabla E(\mathbf{w})^T + \nabla^2 E(\mathbf{w}) \cdot \Delta\mathbf{w}^T, \quad (2.30)$$

wodurch man nach Umstellen die Gewichtsänderung $\Delta\mathbf{w}$ berechnen kann.

$$\Delta\mathbf{w} = -\nabla^2 E(\mathbf{w})^{-1} \cdot \nabla E(\mathbf{w}) \quad (2.31)$$

Es ist ersichtlich, dass die zweite partielle Ableitung der Fehlerfunktion benötigt wird. Für die erste Ableitung der Fehlerfunktion aus Gleichung 2.10 erhält man:

$$\nabla E(\mathbf{w}) = \mathbf{J}(\mathbf{w})^T \cdot (\vec{y}_k - \vec{y}). \quad (2.32)$$

Hierbei ist $\mathbf{J}(\mathbf{w})$ die sogenannte Jacobi-Matrix. Sie enthält die Ableitungen des Ausgabefehlers $\vec{\epsilon}$ nach den Verbindungsgewichten und kann mit Hilfe der Kettenregel sehr einfach über die bereits bekannte Ableitung der Netzausganges nach den Gewichten bestimmt werden.

$$\frac{\partial \vec{\epsilon}}{\partial \mathbf{w}} = \frac{\partial \vec{\epsilon}}{\partial \vec{y}} \cdot \frac{\partial \vec{y}}{\partial \mathbf{w}} = -\frac{\partial \vec{y}}{\partial \mathbf{w}} \quad (2.33)$$

Durch weitere Ableitung der Gleichung 2.32 lässt sich nun die Hesse-Matrix $\mathbf{H}(\mathbf{w})$ berechnen.

$$\nabla^2 E = \mathbf{H}(\mathbf{w}) = \mathbf{J}(\mathbf{w})^T \cdot \mathbf{J}(\mathbf{w}) + \sum_{i=1}^n (y_{k,i} - y_i) \nabla^2 (y_{k,i} - y_i) \quad (2.34)$$

Hierbei bezeichnet der Laufparameter i die Elemente des Ausgabevektors \vec{y} . Das oben beschriebene Newton-Verfahren besitzt eine erhöhte Konvergenzgeschwindigkeit, welche allerdings mit einigen Nachteilen erkauft wird [62], nämlich unter anderem:

- das Verfahren konvergiert nur in der Nähe eines guten Minimums,

- es erfolgt keine Steuerung der Schrittweite und
- der Rechenaufwand zur Bestimmung der Hesse-Matrix ist sehr hoch.

Zur Reduzierung des Rechenaufwandes geht das Gauß-Newton-Verfahren davon aus, dass der Ausgabefehler $\vec{\epsilon}$ gering ist, so dass sich Gleichung 2.34 vereinfacht zu:

$$\mathbf{H}(\mathbf{w}) \approx \mathbf{J}(\mathbf{w})^T \cdot \mathbf{J}(\mathbf{w}) . \quad (2.35)$$

Damit lässt sich Gleichung 2.31 dann wie folgt umformen:

$$\Delta \mathbf{w} = [\mathbf{J}(\mathbf{w})^T \cdot \mathbf{J}(\mathbf{w})]^{-1} \cdot \mathbf{J}(\mathbf{w})^T \cdot (\vec{y}_k - \vec{y}) . \quad (2.36)$$

Um den weiteren Nachteilen des Newton-Verfahrens Rechnung zu tragen, kombiniert der Levenberg-Marquardt-Algorithmus nun den Ansatz aus Gleichung 2.36 mit dem Gradientenabstiegsverfahren aus Abschnitt 2.3.1. Die Gewichtsänderung wird dann mit folgender Vorschrift berechnet:

$$\Delta \mathbf{w} = [\mathbf{J}(\mathbf{w})^T \cdot \mathbf{J}(\mathbf{w}) + \lambda \cdot \mathbf{I}]^{-1} \cdot \mathbf{J}(\mathbf{w})^T \cdot (\vec{y}_k - \vec{y}) . \quad (2.37)$$

Dabei entspricht der Überblendfaktor λ der reziproken Lernrate μ des Gradientenabstiegs, so dass bei großem λ dieser stark überwiegt, während bei kleinem λ das Gauß-Newton-Verfahren überwiegt. Der Überblendfaktor wird dabei bei jedem Lernschritt mit dem Faktor β modifiziert: er wird mit β multipliziert, wenn der Fehler gestiegen ist, um dem Gradientenabstieg stärkeres Gewicht zu verleihen und nach gesunkenem Fehler durch β dividiert, um in einem Minimum die Konvergenzgeschwindigkeit zu erhöhen.

2.4 Universelle Approximationseigenschaften neuronaler Netze

Künstliche neuronale Netze sind in der Lage, nichtlineare funktionale Beziehungen aus den präsentierten Eingangsmustern zu erlernen. Diese mathematische Abbildung lässt sich wie folgt darstellen:

$$f_{NN} : \{X\} \in R^n \rightarrow \{Y\} \in R^m . \quad (2.38)$$

Das neuronale Netz beschreibt demnach den funktionalen Zusammenhang der Eingangsvektoren \vec{x} der Untermenge $\{X\}$ im n-dimensionalen Vektorraum und den Ausgangsvektoren \vec{y} der Untermenge $\{Y\}$ im m-dimensionalen Vektorraum. Darüber hinaus ist in [4] und [5] nachgewiesen worden, dass KNN mit sigmoider Transferfunktion in der verdeckten Schicht jede nichtlineare Funktion mit steigender Anzahl der Neuronen in der verdeckten Schicht beliebig genau approximieren können. Wenn $f(\vec{x}) : R^n \rightarrow R^m$ eine glatte Funktion ist, kann man für jedes kompakte Gebiet $D \subset R^n$ folgenden Zusammenhang schreiben:

$$f(\vec{x}) = \mathbf{w}_*^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}_*^{(1)} \cdot \vec{x}) + \vec{\epsilon}_A \quad \text{mit} \quad \forall \vec{x} \in D \subset R^n . \quad (2.39)$$

Hierbei ist $\vec{\epsilon}_A$ der sogenannte Approximationsfehler, der eine obere Schranke $\bar{\epsilon}$ besitzt und für den $\|\vec{\epsilon}_A\| < \bar{\epsilon}$ gilt. Darüber hinaus ist anzumerken, dass die obere Schranke $\bar{\epsilon}$ mit steigender Anzahl der Neuronen in der verdeckten Schicht sinkt. Die Gewichtsmatrizen $\mathbf{w}_*^{(1)}$ und $\mathbf{w}_*^{(2)}$ stellen die optimale Gewichtungskonfiguration dar, bei der die Schranke $\bar{\epsilon}$ erreicht wird. Daraus lässt sich erkennen, dass das Ziel eines jeden Lernverfahrens sein muss, diese optimalen Gewichtsmatrizen möglichst schnell zu etablieren. Es ist zu unterstreichen, dass die dargestellten universellen Approximationseigenschaften für jede regelungstechnische Anwendung neuronaler Netze große Bedeutung haben, was unter anderem in [16, 45, 47, 52, 63] dokumentiert ist.

2.5 Abbildung dynamischer Prozesse mit neuronalen Netzen

Im Abschnitt 2.4 ist dargestellt worden, dass neuronale Netze generell in der Lage sind, statische nichtlineare Zusammenhänge beliebig genau zu approximieren. Bei nahezu allen realen Prozessen handelt es sich allerdings um nichtlineare dynamische Systeme, bei denen zur Beschreibung des Zusammenhanges zwischen Eingangsgröße \vec{u} und Ausgangsgröße \vec{y} auch der Zustandsvektor \vec{x} und die zeitlichen Abhängigkeiten betrachtet werden müssen. Das zukünftige Verhalten des Systems hängt also vom aktuellen Systemzustand \vec{x} und der Eingangsgröße \vec{u} ab, was sich mit Gleichung 2.40 in allgemeiner Form beschreiben lässt:

$$\begin{aligned}\dot{\vec{x}} &= f(\vec{x}(t), \vec{u}(t)) , \\ \vec{y} &= h(\vec{x}(t)) .\end{aligned}\tag{2.40}$$

Um einen solchen zeitveränderlichen Prozess nun mit einem KNN zu erlernen und abzubilden, ist es nicht ausreichend, nur die Momentanwerte der Ein- und Ausgangsgrößen zu betrachten. Vielmehr ist es nötig, die Ein- und Ausgangswerte des Systems in definierten Zeitfenstern dem Netzwerk zur Verfügung zu stellen [2, 15, 16, 63]. In [15] und [16] ist detailliert dargelegt, dass ein KNN unter der Voraussetzung regelungstechnischer Beobachtbarkeit des abzubildenden Systems die zukünftige Ausgabe $\vec{y}(t+1)$ auf Basis aktueller und vergangener Ein- und Ausgabewerte sicher zu schätzen vermag. Die Größen der dafür nötigen Zeitfenster werden in einem Lernschritt propagiert und zur Gewichts Anpassung genutzt, wodurch die zeitveränderliche Abbildung des Zielsystems erlernt werden kann. Dabei wird im Falle eines Systems mit mehreren Ausgaben $\vec{y}(t)$ in der Regel ein KNN für jede einzelne Ausgabe $y(t)$ vorgesehen. In einem zeitdiskreten Regelungssystem kann dies umgesetzt werden, indem man den Eingabevektor des Netzes $\vec{x}(t)$, wie in Gleichung 2.41 dargestellt, aus den aktuellen Eingangsgrößen $\vec{u}(t)$, deren n zeitverzögerten Werten $\vec{u}(t - n \cdot \Delta t)$ und der Ausgangsgröße $y(t)$ sowie deren n Zeitverzögerungen $y(t - n \cdot \Delta t)$ zusammensetzt.

$$\vec{x} = [\vec{u}(t), \vec{u}(t - \Delta t), \dots, \vec{u}(t - n \cdot \Delta t), y(t), y(t - \Delta t), \dots, y(t - n \cdot \Delta t)]\tag{2.41}$$

Die maximale Anzahl der Verzögerungen und damit die Breite des berücksichtigten Zeitfensters richtet sich nach der Ordnung des abzubildenden Systems [15, 16, 63]. Diese Vorgabe braucht allerdings nicht für jede Anwendung voll ausgeschöpft zu werden, da zum einen die dynamische Abbildung bereits auch mit weniger Eingabewerten gelingen kann; zum anderen wächst mit

einer Vielzahl von Eingabewerten die Netzwerkgröße und damit der Rechenaufwand eines in einem Regelkreis implementierten Netzwerkes. Sind darüber hinaus Größen des Zustandsvektors $\vec{x}(t)$ sensorisch erfassbar, sollte in Erwägung gezogen werden, diese ebenfalls zu nutzen, da sie zusätzliche Informationen über das dynamische Verhalten des Zielsystems beinhalten. In Abbildung 2.4 ist das Blockschaltbild einer Konfiguration dargestellt, mit der das nichtlineare Verhalten eines dynamischen Systems erlernt werden kann.

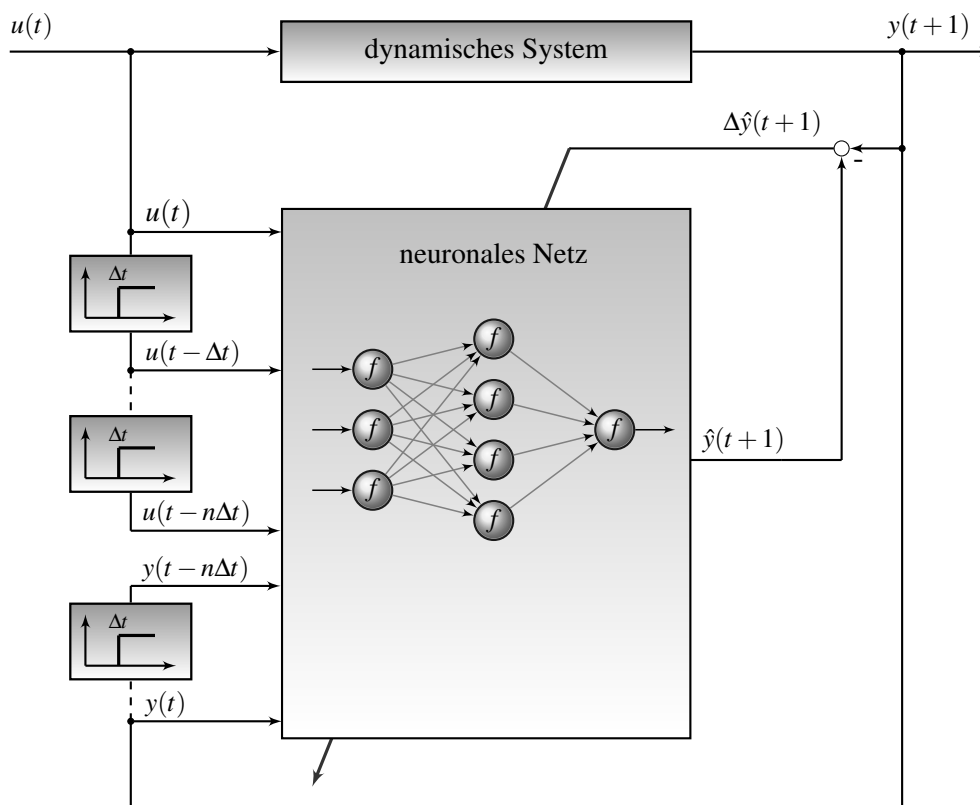


Abbildung 2.4: Abbildung eines dynamischen Prozesses durch ein neuronales Netzwerk.

Hierbei wird der Eingabevektor $\vec{x}(t)$ genutzt, um eine geschätzte Ausgangsgröße $\hat{y}(t+1)$ eines Systems zu ermitteln. Diese wird mit der tatsächlichen Ausgabe des Systems $y(t+1)$ verglichen und deren Differenz $\Delta\hat{y}(t+1)$ dann zum Training des KNN genutzt. Es wird so eine Art Kurzzeitgedächtnis implementiert, indem nicht nur aktuelle Werte, sondern auch historische Daten zur Bestimmung der zukünftigen Ausgangsgröße genutzt werden. Mit diesem generischen Ansatz kann das zeitveränderliche Verhalten nahezu beliebiger beobachtbarer dynamischer Systeme abgebildet und regelungstechnisch nutzbar gemacht werden [15, 16, 64].

2.6 Anwendungsarten neuronaler Lernverfahren

Im Rahmen der regelungstechnischen Nutzung neuronaler Netzwerke lässt sich deren Training unterteilen in das Echtzeit-Lernen im Regelkreis (engl. *online learning*) und das Vortraining (engl. *offline learning*), wobei das Lernen im Regelkreis hierbei hinsichtlich seiner Anwendungshäufigkeit überwiegt. Das Vortraining wird dazu genutzt, einem neuronalen

Regelungselement, bevor es im Regelkreis nutzbar ist, ein bestimmtes Basiswissen aus zuvor gewonnenen Trainingsmustern zu vermitteln. Das Trainingsproblem hierbei und damit die Methoden der praktischen Anwendung verschiedener Lernverfahren sowie die Methoden der Bewertung des Trainingserfolges unterscheiden sich nicht von den klassischen Anwendungen der Regression, Prognose oder Prozessmodellierung [1, 2, 7, 9]. Die verfügbaren Trainingsmuster werden dem KNN dabei normalerweise mehrfach hintereinander in einer bestimmten Anzahl von Lernepochen präsentiert, was den Ausgabefehler meist stark reduziert. In diesem Zusammenhang sei darauf hingewiesen, dass die Begrifflichkeit des *online* und *offline learning* der englischsprachlichen Literatur im regelungstechnischen Kontext nicht eindeutig ist. Der Grund liegt darin, dass beide Begriffe für Arten des Lernens genutzt werden, bei denen das Netzwerk nicht in Echtzeit innerhalb eines geregelten Prozesses trainiert wird.

Der Begriff des *online learning* beschreibt hierbei das sogenannte sequentielle Training, das heißt, die Gewichtsänderung eines KNN erfolgt auf Basis des aktuellen Ausgabefehlers nach jeder Propagation eines Musters. Das Lernen geschieht also in kleinen Schritten, was je nach Qualität der Trainingsdaten und Art der Fehlerfläche zu einem gewissen Rauschverhalten der Gradientenrichtung führen kann [1]. Die Bezeichnung des *offline learning* hingegen meint einen Ansatz der Gewichtsänderung, bei dem die partiellen Gewichtsverschiebungen mehrerer (oder aller) Muster eines Datensatzes gebündelt verarbeitet werden, was mit Gleichung 2.42 verdeutlicht wird.

$$\Delta \mathbf{w} = \Delta_1 \mathbf{w} + \Delta_2 \mathbf{w} + \dots + \Delta_m \mathbf{w} \quad (2.42)$$

Diese Methode wird auch als *batch learning* bezeichnet und beschreibt damit den Ablauf eindeutiger als der Begriff *offline learning*, während es Doppeldeutigkeiten der Namensgebung von Trainingsarten vermeidet. Die akkumulierte Änderung der Verbindungsgewichte aus Gleichung 2.42 bietet den Vorteil, dass eine über m Muster gemittelte Gradientenrichtung ermittelt wird, was ein direkteres Durchschreiten der Fehlerfläche als beim sequentiellen Training ermöglicht. Für diese Art des Lernens wird häufig der Levenberg-Marquardt-Algorithmus aus Abschnitt 2.3.3 verwendet. Der unter anderem in [1] angeführte Nachteil des hohen Rechenaufwandes bei einer großen Anzahl an Trainingsmustern ist mit signifikant gesteigerter Leistungsfähigkeit heutiger Rechner dabei zu vernachlässigen.

Daraus wird deutlich, dass üblicherweise sequentielles Lernen für die Echtzeitanwendung im Regelkreis verwendet wird, während für das Vortraining und ähnliche Trainingsprobleme das *batch*-Verfahren Anwendung findet. Dies liegt nicht zuletzt darin begründet, dass ein adaptives Flugregelungssystem möglichst schnell auf Nichtlinearitäten und Störungen reagieren soll, da das Ausmaß einer Störung meist nicht oder nur sehr schwer zu erfassen ist und ein zu später Trainingsbeginn unter Umständen zu einer kritischen Flugsituation führen kann. Gleichwohl ist eine akkumulierte Gewichtsänderung auf Basis mehrerer aufgezeichneter Zeitschritte für die Echtzeitanwendung nicht komplett ausgeschlossen; es gilt dabei den Vorteil einer direkteren Gradientenrichtung mit dem Aspekt einer verzögerten Reaktion abzuwägen.

2.6.1 Trainingsdatengewinnung und -aufbereitung

Ein neuronales Netzwerk kann nur Zusammenhänge erlernen, die sich auch in hinreichender Weise in den Trainingsdaten widerspiegeln. Soll ein KNN zum Beispiel eine nichtlineare Ausgangsgröße eines Luftfahrzeuges im gesamten Bereich der Flugenveloppe erlernen und abbilden, so wird dies nicht möglich sein, wenn der größte Teil der Trainingsdaten im stationären Reiseflug gewonnen worden ist. Demnach ist für alle Lernprobleme, bei denen ein bestimmter Satz an Trainingsmustern genutzt werden soll, eine gewissenhafte Zusammenstellung und Aufbereitung der Daten für den Lernerfolg unerlässlich. Das Zusammenstellen bezieht sich dabei auf die ingenieurtechnische Analyse und Beurteilung des Informationsgehaltes der Trainingsmuster, während die Aufbereitung den informationstechnischen Aspekt der Daten-normalisierung meint. Da neuronale Netze zum Teil komplexe, nichtlineare Zusammenhänge aus Eingangsgrößen abbilden sollen, die Korrelationen untereinander aufweisen können, ist die Analyse des Informationsgehaltes der Eingaben nicht unbedingt offensichtlich oder leicht interpretierbar. Ein hilfreicher Schritt hierbei ist die Nutzung von Phasendiagrammen, bei denen einzelne potentielle Eingabegrößen eines KNN gegeneinander aufgetragen werden. Häufungen oder große Lücken im Datenraum dieser Größen können so sehr leicht graphisch sichtbar gemacht und physikalisch interpretiert werden. Dadurch kann effektiv beurteilt werden, ob bestimmte Zustände eines Systems in den Trainingsdaten unterrepräsentiert sind und somit von einem KNN nur schlecht erlernt werden können.

Sind die Eingabedaten einer zu erlernenden Abbildung zusammengestellt, ist es für den Lernerfolg nicht unerheblich, die Muster in einen Wertebereich zu transformieren, der vom KNN besser verarbeitet werden kann und somit der Gewichts Anpassung zu Gute kommt. Dafür existieren verschiedene Ansätze der Datenvorverarbeitung, welche die Rohdaten in ein alternatives Format aus Skalenniveau und Verteilung überführen [7]. Eines der gängigsten Verfahren ist die Normalisierung anhand folgender Gleichung [65]:

$$\underline{x}_i^N = \frac{\underline{x}_i - \mu_{x_i}}{k_i \cdot \sigma_{x_i}}. \quad (2.43)$$

Die Berechnung wird für jedes Muster der i -ten Zeile des Eingabevektors $\underline{\vec{x}}$ durchgeführt; dabei ist μ_{x_i} der Erwartungswert und σ_{x_i} die Standardabweichung von \underline{x}_i . Der Faktor k_i legt das Intervall fest, in das die Rohdaten transformiert werden. Die Normalisierung der Ausgabegröße y kann ebenfalls sinnvoll sein, die Berechnung erfolgt analog zu Gleichung 2.43. Im Hinblick auf die netzinterne Signalverarbeitung sollte der Faktor k_i unter Berücksichtigung der verwendeten Transferfunktionen gewählt werden. Werden Tangens Hyperbolicus-Funktionen verwendet, ist es sinnvoll, die Rohdaten in das Zielintervall $[-1; 1]$ zu transformieren, da dadurch verhindert wird, dass die Ausgaben von Neuronen in den asymptotischen Sättigungsbereich laufen und so den Lernprozess behindern.

Für das sequentielle Lernen neuronaler Netze in einem Regelkreis werden Transformationen der Ein- oder Ausgabewerte meist nicht durchgeführt, da die Verteilung der Größen nicht genau

bekannt ist. Dies sollte allerdings nicht grundlegend ausgeschlossen werden, da durch physikalische Systemkenntnis eine Analyse des Signalverhaltens und damit des zu erwartenden Messbereiches durchgeführt werden kann. Durch angemessene Skalierung kann so ebenfalls verhindert werden, dass die Ausgaben von Neuronen mit sigmoiden Transferfunktionen im Sättigungsbereich liegen und damit das Training erschweren.

2.6.2 Bewertung der Generalisierungsfähigkeit

Eine der wichtigsten Eigenschaften eines gut trainierten neuronalen Netzwerkes ist seine Generalisierungsfähigkeit oder auch Robustheit. Darunter versteht man das verallgemeinernde Verhalten eines KNN, auf unbekannte, aber den Trainingsmustern vom Wertebereich ähnliche Eingabedaten mit einer ähnlichen Ausgabe zu reagieren. In Abhängigkeit der Komplexität des Lernproblems steht diese Eigenschaft in direktem Zusammenhang mit der Topologie eines KNN, also der Netzwerkarchitektur und der Art der Transferfunktionen. Ist ein Netz zu klein gewählt, also unterparametrisiert, besitzt es zu wenig Freiheitsgrade, um das Lernproblem angemessen abbilden zu können und liefert große Ausgabefehler. Ist ein Netz hingegen zu groß, verfügt also über zu viele Freiheitsgrade, neigt es zur Überanpassung (engl. *overfitting*), das heißt, es lernt alle präsentierten Trainingsdaten schlichtweg auswendig. Dies hat zur Folge, dass neben dem abzubildenden funktionalen Zusammenhang alle unerwünschten Signaleigenschaften, etwa Messrauschen und statistische Schwankungen, miterlernt werden, was die Generalisierungseigenschaften des KNN stark beeinträchtigt. Ein möglicher Ansatz, die Generalisierungsfähigkeit eines neuronalen Netzes mit einem Qualitätsmaß zu verknüpfen, ist die sogenannte Vapnik-Chervonenkis-Dimension [1, 2]. Dieses statistisch abgeschätzte Maß der Abbildungskapazität eines KNN ist allerdings nicht allgemein gültig, sondern muss in Abhängigkeit von der Topologie für jedes Netzwerk in bestimmten Intervallgrenzen ermittelt werden [9], was einen nicht unerheblichen Aufwand mit sich bringt. Eine genaue Abwägung von Aufwand und Nutzen ist diesbezüglich also ratsam.

Eine andere Möglichkeit, die Generalisierungseigenschaften zu bewerten, ist die sogenannte (Kreuz-)Validierung (engl. *cross validation*) [1, 2, 7]. Bei diesem Validierungsansatz werden üblicherweise 10-15 % aller verfügbaren Trainingsmuster zufällig ausgewählt und dem KNN für die Gewichts Anpassung vorenthalten. Diese sind dem Netzwerk somit unbekannt, bewegen sich aber im gleichen Wertebereich wie die Trainingsdaten. Werden diese Daten nun dem KNN präsentiert, kann man über den resultierenden Validierungsfehler eine Aussage über die Robustheit treffen, was in Abbildung 2.5 verdeutlicht wird.

Die Abbildung zeigt, dass der Trainingsfehler üblicherweise mit der Anzahl der Epochen stetig abnimmt, während es für den Validierungsfehler häufig einen optimalen Punkt gibt, nach dessen Überschreiten der Fehler wieder ansteigt. Ein steigender Validierungsfehler bei sinkendem Trainingsfehler ist ein klares Indiz für beginnendes Auswendiglernen unerwünschter Zufallsschwankungen. Um die Generalisierungseigenschaft eines KNN nicht zu reduzieren, ist es ratsam, an einem solchen Punkt das Training vorzeitig zu beenden (engl. *early stopping*). Das

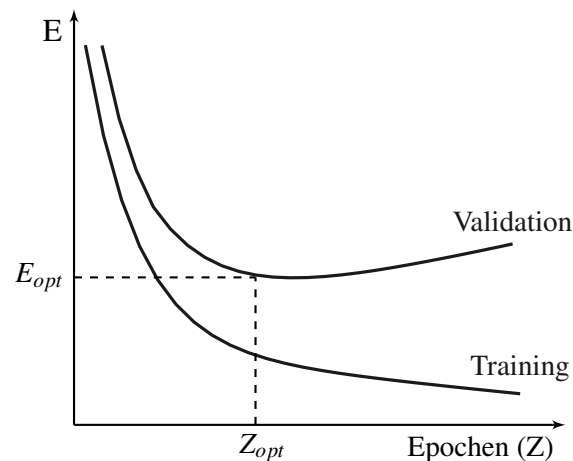


Abbildung 2.5: Bewertung der Generalisierungsfähigkeit anhand des Validationsfehlers.

Bewertungsmaß hierfür ist üblicherweise ein über mehrere Epochen hintereinander ansteigender Validierungsfehler. Tritt ein solcher Verlauf des Validierungsfehlers auf, sollte auch geprüft werden, ob das trainierte Netzwerk gegebenenfalls über zu viele Freiheitsgrade verfügt, es also ein Übermaß an Gewichten aufweist.

2.7 Systematischer Entwurfsprozess für neuronale Netze

Aus den bisherigen Ausführungen wird deutlich, dass der Entwurf eines neuronalen Netzwerkes für ein bestimmtes Lernproblem einen iterativen Prozess mit vielen Einflussparametern darstellt. Diese sind unter anderem die Anzahl der Schichten und ihrer Neuronen, der Grad der Vernetzung mit Gewichten und die Nutzung von Rückkopplungen und Abkürzungen sowie die Wahl der Transferfunktion und des passenden Lernverfahrens. Dabei stellt sich nicht nur die Frage des Algorithmus' (siehe Abschnitt 2.3), sondern auch, ob die Gewichtsänderungen sequentiell oder gebündelt erfolgen sollen. Zur Bestimmung der notwendigen Neuronenzahl für eine bestimmte Trainingsaufgabe existieren zwar Näherungssätze [9], allerdings gehen diese von einem festen Satz Trainingsmuster aus, was für die regelungstechnische Anwendung meist nicht zielführend ist. Für jede Anwendung eines KNN zur Abbildung eines komplexen nichtlinearen Zusammenhanges kann es also nicht das Ziel sein, ein Netzwerk mit all seinen Parametern so lange zu optimieren, bis es die gewünschte Aufgabe adäquat erfüllt. Vielmehr ist ein systematischer und ingenieurtechnisch reproduzierbarer Entwurfsprozess wünschenswert, wie er in Abbildung 2.6 dargestellt ist. Die Besonderheit dieses Ansatzes ist, dass nicht nur ein Netzwerk entworfen und bewertet wird, sondern dass große Gruppen von Netzwerken (z. B. 100 oder mehr) für eine Lernaufgabe trainiert und ihr Lernerfolg statistisch über alle Netze analysiert werden kann. Dies hat den Vorteil, dass zufällige Schwankungen, etwa durch die Initialisierung der Verbindungsgewichte, ausgeglichen werden und man zu einer allgemeinen Aussage kommen kann, ob das gewünschte Modell durch neuronale Netze abgebildet werden kann. In [66] ist ein Ansatz zur Auslegung vortrainierter neuronaler Netzwerke für sicherheitskritische Anwendungen in der Luft- und Raumfahrt dargestellt, der gezielt Konzepte der Verifikation und

Validation (V&V) berücksichtigt. Dabei werden auch Zertifizierungsrichtlinien für Software in Luftfahrtanwendungen wie die DO-178B einbezogen. Ohne den Anspruch auf Zertifizierbarkeit zu erheben, zeigt sich gleichwohl, dass der in Abbildung 2.6 beschriebene Ansatz deutliche Parallelen zu dem V&V-Prozess aus [66] aufweist.

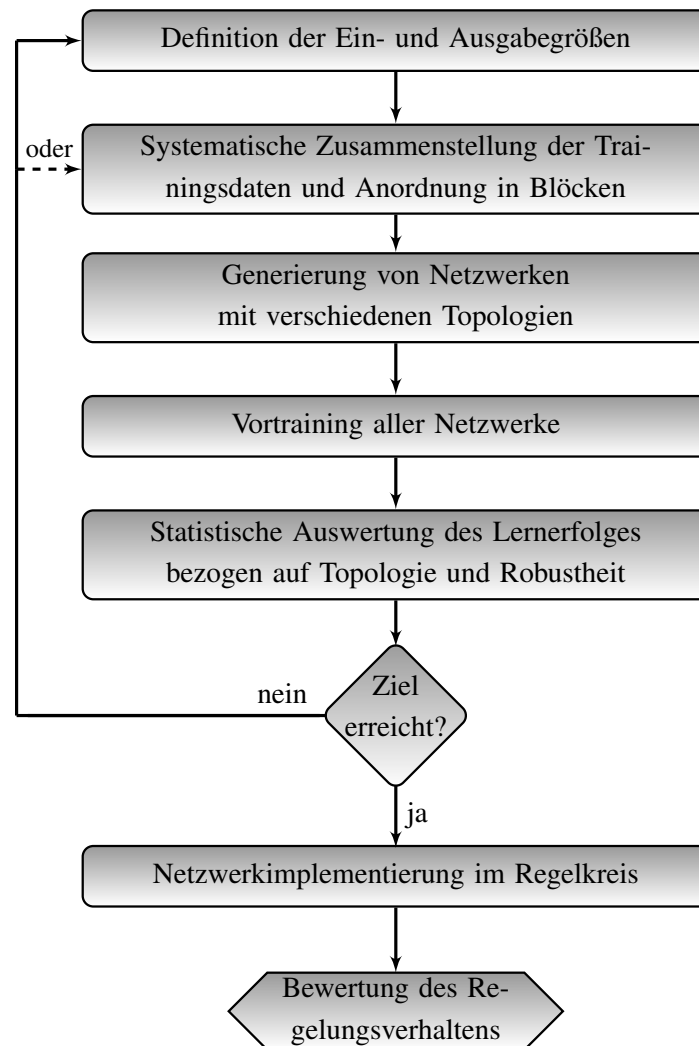


Abbildung 2.6: Systematischer Entwurf und Bewertung neuronaler Netzwerke.

Um eine bestimmte Zielgröße mit einem KNN abbilden zu können, ist die Auswahl potentieller Eingabesignale von großer Bedeutung; dies gilt sowohl hinsichtlich der Darstellung dynamischen Verhaltens gemäß Abschnitt 2.5 als auch in Bezug auf den Informationsgehalt der Trainingsdaten (siehe Abschnitt 2.6.1). Gerade bei diesem ersten Schritt des Entwurfes ist eine gewisse Systemkenntnis hinsichtlich der Prozessgrößen und ihrer Einflüsse von Vorteil, da die falsche Wahl der Eingangssignale den Lernprozess erschweren oder gar unmöglich machen kann. Sind die Netzeingaben definiert, können im nächsten Schritt die Trainingsmuster zusammengestellt und normiert werden. Hinsichtlich des Informationsgehaltes der Trainingsdaten kann es zweckmäßig sein, diese in verschiedene Blöcke oder nach Systemzuständen zu unterteilen. Im Falle eines Flugzeuges kann dies beispielsweise bedeuten, dass die Trai-

ningsdaten nach verschiedenen Flugzuständen oder Windverhältnissen unterteilt sind. Dadurch wird die Menge der Trainingsmuster in kleinere, klar abgegrenzte Bereiche aufgeteilt, was sie physikalisch leichter interpretierbar macht (siehe Abschnitt 2.6.1). Außerdem kann der Lernerfolg im weiteren Verlauf auch hinsichtlich einzelner Datenblöcke analysiert werden, was gegebenenfalls zusätzliche Informationen auf Über- oder Unterrepräsentierung bestimmter Systemzustände in den Trainingsmustern liefert.

Wie bereits ausgeführt, werden im Vortraining große Gruppen von Netzen auf ein Lernproblem trainiert. Um dem Aspekt zu vieler oder zu weniger Freiheitsgrade für eine gute Generalisierungsfähigkeit gemäß Abschnitt 2.6.2 Rechnung zu tragen, ist es sinnvoll, sich dabei verschiedener Topologien zu bedienen. Für vorwärtsgerichtete Netzwerke bedeutet dies, die Anzahl der verdeckten Schichten, respektive die Anzahl Neuronen in diesen, deutlich zu variieren. Im Falle der besagten 100 Netzwerke für ein Lernproblem kann man sich zum Beispiel Gruppen zu 10 oder 20 Netzwerken einer Topologie vorstellen. Dadurch, dass die Verbindungsgewichte aller Netzwerke zufällig initialisiert werden, kann durch die statistische Bewertung der einzelnen Gruppen die Eignung einer bestimmten Topologie klarer bewertet werden.

Das Training einer großen Menge von Netzwerken und dessen Auswertung beinhaltet möglichst viele Zufallsentscheidungen, die die statistische Auswertung des Lernerfolges verlässlicher machen. Neben der zufälligen Initialisierung der Gewichte aller Netzwerke wird auch die Auswahl von Trainings-, Test- und Validierungsmustern für jedes Netzwerk in jeder Lernepoche zufällig durchgeführt. Testdaten unterscheiden sich dabei nur insofern von den Validierungsdaten, als dass mit ihnen kein vorzeitiger Trainingsabbruch bei steigendem Fehler über mehrere Epochen verknüpft ist. Das bedeutet, dass nach einem vorzeitigen Trainingsabbruch die Testdaten dem KNN präsentiert werden können, um eine angemessene Generalisierungsfähigkeit zu bestätigen oder die Topologie als potentiell überangepasst zu identifizieren. Die Aufteilung erfolgt häufig in einem Verhältnis von 70 % Trainingsdaten und jeweils 15 % Test- und Validierungsdaten; mit einem Anteil von 30 % in jeder Epoche zufällig ausgewählter und dem Netz im Training vorenthaltener Daten kann somit eine fundierte Aussage über die verallgemeinernden Eigenschaften einer Topologie gemacht werden. Dies gilt besonders, da die Menge aller Netze häufig über mehrere Dutzend oder hundert Epochen trainiert wird und so eine breite Datenbasis zur Bewertung des Lernerfolges zur Verfügung steht. Da die Test- und Validierungsmuster aus allen zuvor generierten Trainingsdatenblöcken verschiedener Systemzustände gleichmäßig gewonnen werden, kann die Robustheit auch für einzelne Datenblöcke betrachtet werden.

Im Rahmen der Analyse des Lernerfolges kann man schließlich auch Fehlerbetrachtungen über alle Netzwerke einer Topologie oder alle trainierten Netze durchführen. Die Betrachtung aller Netzwerke gibt häufig einen schnellen und zuverlässigen Eindruck, ob das Lernproblem angemessen abgebildet werden konnte. Die Analyse besonders der Test- und Validierungsdaten

aller Topologien lässt wiederum häufig einen schnellen Schluss zu, welche Netzwerkgröße der Lernaufgabe angemessen ist, da der Validationsfehler in diesen Fällen ein Minimum aufweist. Sollte darüber hinaus weiterer Bedarf bestehen, die Robustheit der trainierten Netzwerke zu prüfen, können zum Beispiel mit dem sogenannten *bootstrap*-Verfahren zusätzliche Testdaten generiert werden [1]. Bei diesem Verfahren werden auf Basis der statistischen Verteilung der Originaldaten neue ähnliche Eingangsmuster erzeugt, die dem KNN präsentiert werden können. Es ist auch denkbar, dass durch die Auswertung des Lernerfolges der Eindruck entsteht, dass bestimmte Eingabegrößen trotz bewusster Auswahl kaum zur Modellbildung beitragen. In einem solchen Fall bietet sich eine nachträgliche Optimierung des Vektors der Eingabegrößen hinsichtlich des Informationsgehaltes an. Dies lässt sich vollständig automatisiert und in zuverlässiger Weise mit den zuvor erwähnten Verfahren der evolutionären Algorithmen oder der Partikelschwarmoptimierung durchführen. Die damit unter Umständen verbundene Reduzierung der Eingabesignale hat auch eine Verkleinerung des KNN zur Folge, was im Sinne des Rechenaufwandes für die Echtzeitanwendung von Interesse ist.

Nach endgültiger Bewertung der Lernphase stehen sowohl über alle Netze betrachtet, als auch für die unterschiedlichen Topologien oder einzelnen Netzwerke ausreichend Informationen zur Verfügung, um Kandidatennetze für die Anwendung im Regelkreis auszuwählen. Es sei an dieser Stelle betont, dass der Entwurfsprozess aus Abbildung 2.6 generisch und damit nicht auf regelungstechnische Anwendungen beschränkt ist. Gleichwohl ist diese Systematik ein wichtiger Teilaspekt für Anwendung neuronaler Netze in der Regelung, da durch die beschriebene statistische Auswertung der Trainingsverlauf reproduzierbar wird. Dem häufig geäußerten Kritikpunkt, neuronale Netzwerke seien in ihrer Arbeitsweise allgemein nicht nachvollziehbar, kann mit diesem Entwurfsprozess von statistischer Seite somit widersprochen werden.

3 Anwendung der strukturvariablen Regelung auf neuronale Lernverfahren

Bei der Regelung komplexer nichtlinearer Prozesse, wie zum Beispiel der Flugzeugsteuerung, kommt es häufig zu der ingenieurstechnischen Fragestellung inwieweit einzelne Teilaspekte, unter Umständen durch vereinfachende Annahmen wie der Linearisierung, analytisch beschrieben werden können. In diesem Zusammenhang ist es naheliegend, dass die Lernfähigkeit und die Approximationseigenschaften neuronaler Netzwerke einen erheblichen regelungstechnischen Mehrwert darstellen können. Gleichwohl werden neuronale Netze in der konventionellen Regelungstechnik trotz systematischer Auslegung häufig als *black box* bezeichnet, womit die komplexe analytische Beschreibbarkeit der netzinternen Prozesse bei sich ändernden Gewichtsmatrizen charakterisiert werden soll. Trotz der Möglichkeit, mit der Ljapunow'schen Theorie global asymptotische Stabilität neuronal beeinflusster Regelsysteme zu beweisen, werden derartige Regelungsansätze bei sicherheitskritischen Anwendungen, wie etwa der Flugsteuerung, nur mit reduzierter Autorität in Erwägung gezogen [28]. Unter diesen Gesichtspunkten erscheint es zweckmäßig, die Sichtweise auf neuronale Netzwerke zu erweitern und ihren Lernprozess als regelungstechnischen Prozess zu betrachten. Eine Möglichkeit hierfür ist die sogenannte strukturvariable Regelung, respektive ihre Sonderform der Gleitzustandsregelung, die in der Vergangenheit bereits auf neuronale Netze übertragen wurde [25, 26, 67, 68]. Diese Lernverfahren sind zum einen robust gegenüber externen Störungen und zum anderen verfügen sie wegen ihrer dynamisch bestimmbaren Lernrate meist über eine höhere Konvergenzgeschwindigkeit als die in Kapitel 2 beschriebenen Standardansätze.

Strukturvariable Regelungssysteme (engl. *variable structure systems*) können, wie der Name impliziert, in Abhängigkeit der Zustandsgrößen eines zu regelnden Systems zwischen einzelnen Stellgesetzen umschalten, also ihre Struktur zwischen verschiedenen Reglern verändern. Sie werden zur Regelung linearer wie auch nichtlinearer Systeme verwendet und wurden ab den 1950er Jahren zunächst maßgeblich in der damaligen Sowjetunion entwickelt [69]. Die besagte Strukturveränderung erfolgt in Abhängigkeit der Zustände abrupt und führt demnach dazu, dass diese Klasse von Regelungssystemen über ein diskontinuierliches Stellgesetz verfügt [70, 71]. Damit die Umschaltung zwischen verschiedenen Stellgesetzen systematisch erfolgen kann, werden dazu geeignete Schaltfunktionen definiert, die den Zustandsraum des betrachteten Systems in einzelne, den Stellgesetzen zugeordnete Bereiche unterteilen. Ein so aus mehreren Subsystemen/Reglern synthetisiertes strukturumschaltendes Gesamtsystem kann dabei gänzlich neue dynamische Eigenschaften aufweisen, die keinem der Teilsysteme zu eigen sind. Bei geeigne-

ten Schaltfunktionen ist es sogar möglich, aus für sich instabilen Subsystemen ein global asymptotisch stabiles Regelungssystem zu realisieren, wie in [70–73] dargelegt. Unter bestimmten Umständen ist es zusätzlich möglich, dass es zu einer besonderen Dynamik der Systemzustände kommen kann, dem sogenannten Gleitzustand (engl. *sliding mode*). Ein erreichter Gleitzustand bietet zwei erhebliche regelungstechnische Vorteile: er ist von Natur aus global asymptotisch stabil und weist gleichzeitig ein hohes Maß an Robustheit gegenüber Störungen auf.

3.1 Grundbegriffe der strukturvariablen Regelung

Bevor der Ansatz der strukturvariablen Regelung und besonders des Gleitzustandes auf neuronale Netzwerke übertragen werden kann, sollen zunächst einige zentrale Grundlagen ihrer Theorie dargelegt werden. Es ist bereits angemerkt worden, dass dieser Regelungsansatz für nichtlineare Systeme, wie sie in Gleichung 2.40 beschrieben sind, genutzt werden kann. Da neuronale Netze ebenfalls nichtlineares Verhalten aufweisen, ist dies eine wichtige Grundvoraussetzung. Geht man weiterhin davon aus, dass das System linear bezüglich seiner Steuerung \vec{u} ist, so kann man folgende allgemeine Beschreibung für ein System n -ter Ordnung finden:

$$\begin{aligned}\dot{\vec{x}} &= \vec{f}(\vec{x}, t) + \mathbf{B}(\vec{x}, t) \cdot \vec{u}, \\ \vec{y} &= \vec{h}(\vec{x}, t).\end{aligned}\tag{3.1}$$

Hierbei sind $\vec{x} = [x_1, x_2, \dots, x_n]^T$ der Zustandsvektor, $\vec{u} = [u_1, u_2, \dots, u_m]^T$ der Eingangsvektor der Länge m , $\vec{y} = [y_1, y_2, \dots, y_m]^T$ der Ausgangsvektor und $\mathbf{B}_{n \times m}(\vec{x}, t)$ die Eingangsmatrix des Systems. Wie bereits erwähnt, besitzen alle strukturumschaltenden Systeme ein vom Zustandsvektor \vec{x} abhängiges, diskontinuierliches Stellgesetz:

$$\vec{u}(\vec{x}) = \begin{cases} \vec{u}^+(\vec{x}, t) & \text{für } \vec{S}(\vec{x}) > 0 \\ \vec{u}^-(\vec{x}, t) & \text{für } \vec{S}(\vec{x}) < 0 \end{cases}.\tag{3.2}$$

An dieser Stelle wird mit $\vec{S}(\vec{x}) = [S_1(\vec{x}), S_2(\vec{x}), \dots, S_m(\vec{x})]^T$ der Vektor der m Schaltfunktionen eingeführt, die das Umschalten zwischen den einzelnen kontinuierlichen Regelgesetzen \vec{u}^+ und \vec{u}^- in Abhängigkeit der Position im Zustandsraum vorgeben. Je nach Ordnung des Systems handelt es sich bei den Schaltfunktionen um sogenannte Schaltlinien, -flächen oder Schalthyperflächen; die Gesamtheit aller Schaltfunktionen wird auch Schaltmannigfaltigkeit genannt (engl. *switching manifold*). Da in dieser Arbeit im Sinne der Systemordnung nur Schaltlinien zum Einsatz kommen, soll dieser Begriff im weiteren Verlauf verwendet werden. Mit den getroffenen Aussagen wird deutlich, dass die Schaltfunktionen und die Stellgesetze die Auslegungsparameter eines strukturvariablen Regelungssystems darstellen. An dieser Stelle sei außerdem angemerkt, dass sich die Nomenklatur der Gleichungen 3.1 und 3.2 an der verbreiteten Schreibweise aus [25, 69, 70] und [74] orientiert. Es ist auch gängig, die Funktion \vec{f} durch die Dynamikmatrix \mathbf{A} zu ersetzen [73, 75].

Zuvor ist bereits ausgeführt worden, dass ein aus geeigneten Subreglern zusammengesetztes Gesamtsystem neue dynamische Eigenschaften aufweisen kann. Bei entsprechender Auslegung

können positive Eigenschaften von Subsystemen kombiniert werden, während negative entfallen. Was dies konkret heißt, soll qualitativ an einem System zweiter Ordnung erläutert werden, das aus zwei grenzstabilen Teilsystemen besteht, zwischen denen umgeschaltet wird. Das Phasendiagramm dieses Reglers ist in Abbildung 3.1 dargestellt, wobei eine detaillierte analytische Beschreibung in [72] gegeben ist. Nach [69] stellt diese Regelung das erste überhaupt veröffentlichte Beispiel eines strukturvariablen Systems dar.

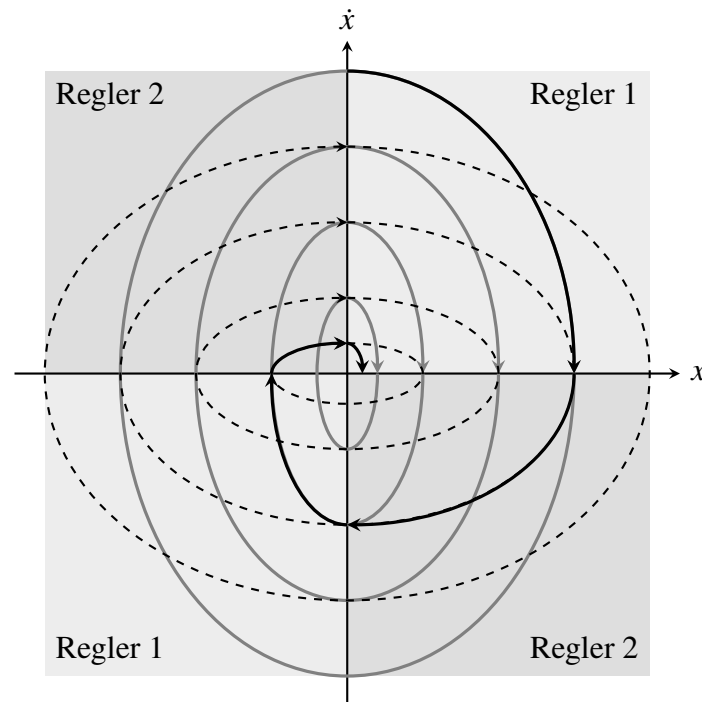


Abbildung 3.1: Prinzipielle Darstellung der Strukturumschaltung zwischen zwei Reglern.

Es ist erkennbar, dass die Trajektorien des Teilsystems des Reglers 1 aus einer Schar aufrecht „stehender“ Ellipsen bestehen, während das System des Reglers 2 aus einer ähnlichen Schar „liegender“ Ellipsen besteht; beide Teilsysteme sind für sich genommen augenscheinlich grenzstabil. Die zwei Schaltfunktionen verlaufen entlang der Abszisse und der Ordinate, so dass die vier Quadranten des Koordinatensystems die Gültigkeit der jeweiligen Stellgesetze markieren. Man sieht, dass auf der Abszisse zwischen Quadrant eins und vier zwischen Regler 1 und Regler 2 umgeschaltet wird, so dass die Zustände entsprechend geänderten Trajektorien folgen. Gleiches geschieht beim Wechsel von Quadrant vier nach drei und erneut von Quadrant drei nach zwei. Es wird deutlich, dass die Zustände durch das Umschalten immer näher an den stabilen Koordinatenursprung geführt werden und dass tatsächlich aus zwei grenzstabilen Systemen ein global asymptotisch stabiles Gesamtsystem synthetisiert werden kann [72].

Eine besondere Form der strukturvariablen Regelung ist jene mit auftretendem Gleitzustand, auch Gleitzustandsregelung (engl. *sliding mode control*) genannt. Dabei handelt es sich um eine erzwungene, global asymptotisch stabile Systemdynamik, die auftritt, wenn sich die Zustands-

größen direkt auf einer Schaltlinie bei $S = 0$ befinden (bei höheren Ordnungen befinden sich die Zustände auf einer Schaltfläche/-hyperfläche oder auf einer Schnittgeraden entsprechender Flächen und es gilt $\vec{S} = 0$). Im Gleitzustand wird eine Schaltlinie folgerichtig als Gleitlinie bezeichnet. Dabei ist festzustellen, dass der Gleitzustand im Gegensatz zu Abbildung 3.1 nur auftreten kann, wenn die Trajektorien der Zustandsgrößen im Phasenraum aus entgegengesetzter Richtung der Schaltlinie zustreben. Daran wird auch deutlich, dass ein System, sobald es den Gleitzustand erreicht hat, in diesem verharnt, also $\dot{S} = 0$ gilt. Dies wird dadurch sichergestellt, dass, sobald die Zustände die Gleitlinie in eine Richtung auch nur minimal verlassen, das diskontinuierliche Stellgesetz aus Gleichung 3.2 diese wieder zurück auf die Gleitlinie zwingt. Ein System im Gleitzustand verbleibt also in demselben und gleitet dem stabilen Koordinatenursprung des Zustandsraumes entgegen. Die Dynamik, mit der dies geschieht, wird dabei durch die Mannigfaltigkeit der Schaltfunktionen vorgegeben und ist folglich Gegenstand der ingenieurtechnischen Auslegung. Durch diese Verknüpfung von Subsystemen kann eine Dynamik realisiert werden, die von den einzelnen Teilsystemen entkoppelt ist und ein hohes Maß an Robustheit gegenüber Störungen und sogar Veränderungen der Streckenparameter aufweist [69–75]. Zur graphischen Verdeutlichung ist in Abbildung 3.2 das oben beschriebene Prinzip für ein System zweiter Ordnung mit zwei Schaltlinien, von der eine den Gleitzustand aufweist, qualitativ dargestellt.

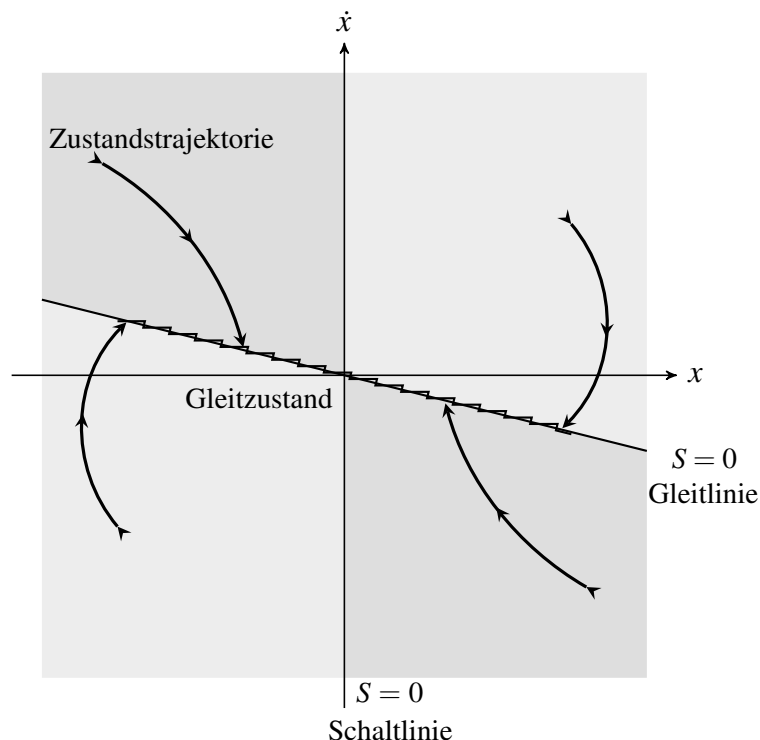


Abbildung 3.2: Erreichen des Gleitzustandes bei einem System zweiter Ordnung.

Soll nun die Dynamik im Gleitzustand entwurfstechnisch vorgegeben werden, so muss man bei genauer Betrachtung der Gleichungen 3.1 und 3.2 feststellen, dass das System wegen des diskontinuierlichen Stellgesetzes an der Stelle $S = 0$ bisher nicht beschrieben ist. Demzufolge kann

auch kein Stellsignal ermittelt werden, welches sicherstellt, dass die Zustände mit der Bedingung $\dot{S} = 0$ in Richtung des Phasenraumursprungs gleiten. Es existieren verschiedene Ansätze zur Beschreibung der Dynamik im Gleitzustand wie zum Beispiel die Zustandstransformation oder die Methode nach Filippov [70, 73, 74]. Ein weniger abstrakter Ansatz ist die sogenannte äquivalente Regelung (engl. *equivalent control*). Die Grundidee besteht darin, die Stellgröße \vec{u} im Falle des Gleitzustandes durch eine kontinuierliche äquivalente Stellgröße \vec{u}_{eq} zu ersetzen, wobei im Gleitzustand das äquivalente dieselbe Dynamik beschreibt wie das Ursprungssystem [25, 70, 71, 73, 74]. Allgemein formuliert gilt es also, die Stellgröße \vec{u}_{eq} für den Fall, dass $\dot{S} = 0$ ist, zu finden. Dazu wird zunächst mit Hilfe der Kettenregel eine Umformung durchgeführt:

$$\dot{S}(\vec{x}) = \frac{\partial \vec{S}}{\partial \vec{x}} \cdot \frac{\partial \vec{x}}{\partial t} = \frac{\partial \vec{S}}{\partial \vec{x}} \cdot \dot{\vec{x}} = 0. \quad (3.3)$$

Die partiellen Ableitungen $\partial \vec{S} / \partial \vec{x}$ werden hierbei auch häufig mit der Matrix \mathbf{G} beschrieben. In obigen Zusammenhang kann nun Gleichung 3.1 eingesetzt werden:

$$\dot{S}(\vec{x}) = \frac{\partial \vec{S}}{\partial \vec{x}} \cdot [\vec{f}(\vec{x}, t) + \mathbf{B}(\vec{x}, t) \cdot \vec{u}] = 0. \quad (3.4)$$

Löst man dies nun nach \vec{u} auf und setzt \vec{u}_{eq} ein, so erhält man das Stellgesetz der äquivalenten Regelung gemäß Gleichung 3.5.

$$\vec{u}_{eq} = - \left(\frac{\partial \vec{S}}{\partial \vec{x}} \cdot \mathbf{B}(\vec{x}, t) \right)^{-1} \cdot \frac{\partial \vec{S}}{\partial \vec{x}} \cdot \vec{f}(\vec{x}, t) \quad (3.5)$$

Hieran wird deutlich, dass das Produkt aus Eingangsmatrix $\mathbf{B}(\vec{x}, t)$ und Gradientenmatrix $\partial \vec{S} / \partial \vec{x}$ invertierbar sein muss. Setzt man nun wiederum die äquivalente Stellgröße \vec{u}_{eq} in Gleichung 3.1 ein, so erhält man folgenden Ausdruck:

$$\dot{\vec{x}} = \vec{f}(\vec{x}, t) - \mathbf{B}(\vec{x}, t) \cdot \left(\frac{\partial \vec{S}}{\partial \vec{x}} \cdot \mathbf{B}(\vec{x}, t) \right)^{-1} \cdot \frac{\partial \vec{S}}{\partial \vec{x}} \cdot \vec{f}(\vec{x}, t). \quad (3.6)$$

Damit ist mit Hilfe der äquivalenten Regelung die Dynamik im Gleitzustand beschrieben und demzufolge auch ihre Eindeutigkeit gewährleistet. Darüber hinaus ergibt sich nach [71] eine interessante physikalische Interpretation dieser Zusammenhänge. Demnach besteht der Gesamtstellvektor aus zwei Anteilen: nämlich einem, der die Zustände trotz Störungen auf der Gleitlinie hält und einem weiteren, der äquivalenten Stellgröße, die die Zustände in Richtung des stabilen Koordinatenursprungs gleiten lässt.

3.2 Stabilitätsbetrachtung nichtlinearer Systeme

Bevor auf weitere Auslegungsaspekte der Gleitzustandsregelung eingegangen wird, soll an dieser Stelle in kurzer Form die Ljapunow'sche Theorie dargestellt werden, mit der die (regelungstechnische) Stabilität nichtlinearer Systeme untersucht werden kann. Der Vorteil der sogenannten direkten Methode, die Ende des 19. Jahrhunderts von Alexander Ljapunow

entwickelt wurde, besteht darin, dass eine Aussage über die Stabilität eines nichtlinearen Systems möglich ist, ohne dessen Differentialgleichungen lösen zu brauchen. Umfangreiche Darstellungen dieser Stabilitätstheorie sind in [76, 77] gegeben, während in [72, 78] die regelungstechnische Anwendung zusammengefasst ist und zum Beispiel in [46] ein direkter Bezug zu adaptiven Flugregelungssystemen hergestellt wird. Allgemein lässt sich feststellen, dass die direkte Methode von Ljapunow Standard für die Stabilitätsanalyse nichtlinearer Systeme ist.

Die Grundidee der direkten Methode besteht vereinfacht gesagt in einer Energiebetrachtung des zu untersuchenden Systems. Nimmt man als Beispiel ein physikalisches System, so ist seine Ruhelage erreicht, wenn kinetische und potentielle Energie gleich Null sind. Außerdem ist es nachvollziehbar, dass die Energie eines stabilen Systems zu keinem Zeitpunkt ohne äußere Anregung zunehmen darf. Wenn es gelingt, eine Ljapunow-Funktion $V(\vec{x})$ zu finden, die die Energie des Systems in Abhängigkeit der Zustandsgrößen \vec{x} beschreibt, und man nachweisen kann, dass die Energie stetig abnimmt, so ist eine Stabilitätsaussage ohne explizite Lösung der Systemgleichungen möglich. Graphisch ist dies in Abbildung 3.3 verdeutlicht. Darin sind die Höhenlinien einer trichterförmigen Ljapunow-Funktion dargestellt, die bildlich gesprochen das Energieniveau des Systems charakterisieren. Die Stabilität des betrachteten Systems ist offenbar gewährleistet, wenn sich die Trajektorie der Zustandsgrößen die Höhenlinien kreuzend in Richtung Koordinatenursprung bewegt.

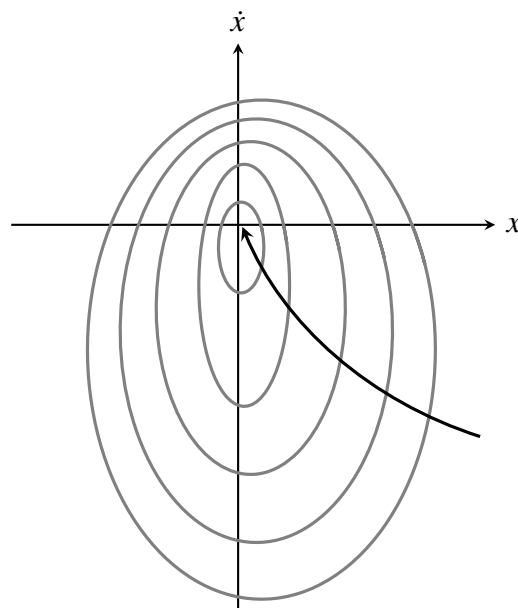


Abbildung 3.3: Asymptotische Stabilität am Beispiel von Höhenlinien einer Ljapunow-Funktion.

Mathematisch lassen sich klare Forderungen an eine Funktion $V(\vec{x})$ ableiten, mit denen verschiedene Abstufungen der Ljapunow-Stabilität definiert werden können [46, 72, 76–78]. Diese Abstufungen sind die einfache Stabilität, die asymptotische und die global asymptotische, auch vollständige Stabilität genannt. Ein System ist einfach stabil, wenn eine betrachtete Ruhelage

\vec{x}_0 stabil ist, die Zustände also mit fortschreitender Zeit stets in ihrer Umgebung Ω verbleiben. Zusätzlich liegt asymptotische Stabilität vor, wenn die Zustände bei fortschreitender Zeit in Richtung der Ruhelage streben. Ein System ist global asymptotisch stabil, wenn die asymptotische Stabilität nicht nur für eine Umgebung um die Ruhelage, sondern für den gesamten Zustandsraum gilt. Mathematisch lässt sich dies folgendermaßen zusammenfassen:

1. Einfache Stabilität für einen betrachteten Bereich mit begrenztem Ω um \vec{x}_0 gilt, wenn:

- $V(\vec{x})$ stetig differenzierbar ist,
- $V(\vec{x} = 0) = 0$,
- $V(\vec{x}) > 0 \quad \forall \vec{x} \in \Omega \text{ und } \vec{x} \neq 0$,
- $\dot{V}(\vec{x}) \leq 0 \quad \forall \vec{x} \in \Omega$.

2. Asymptotische Stabilität ist gegeben, wenn zusätzlich zu 1. gilt:

- $\dot{V}(\vec{x}) < 0 \quad \forall \vec{x} \in \Omega$,
- $\dot{V}(\vec{x}) = 0 \quad \text{für } \vec{x} = 0$.

3. Global asymptotische Stabilität mit unbegrenztem Ω liegt vor, wenn zusätzlich gilt:

- $\lim_{\|\vec{x}\| \rightarrow \infty} V(\vec{x}) = \infty$.

Wird ein Regelkreis untersucht, handelt es sich dabei um ein zeitdiskretes System, so dass man die obige Betrachtungsweise kontinuierlicher Systeme für einzelne Zeitschritte t wie folgt abwandeln kann:

$$\dot{V}(\vec{x}) = \Delta V = V(t+1) - V(t) < 0. \quad (3.7)$$

Aus diesen Ausführungen wird deutlich, dass für die regelungstechnische Anwendung möglichst global asymptotische Stabilität nachzuweisen ist. Bei komplexen Systemen besteht die Herausforderung hierbei darin, eine geeignete Ljapunow-Funktion $V(\vec{x})$ zu finden, die das Systemverhalten adäquat beschreibt [72, 77, 78].

3.3 Systeme im Gleitzustand

Da durch die Einführung der äquivalenten Regelung die Dynamik im Gleitzustand eindeutig beschrieben werden kann, ist es ebenfalls möglich, diese Dynamik durch die Definition der Schaltfunktionen vorzugeben. Soll also ein Gleitzustandsregler ausgelegt werden, so ergeben sich nach [70] stets zwei aufeinander folgende Entwurfsschritte:

- Die gewünschte Dynamik des Gleitzustandes wird über die Definition der Schaltfunktionen vorgegeben.
- Es muss ein diskontinuierliches Schaltgesetz bestimmt werden, das einen stabilen Gleitzustand ermöglicht.

Betrachtet man nun ein System n -ter Ordnung mit m Eingangsgrößen, so kann man zeigen, dass in $2^m - 1$ Schnittmengen der m Schalthyperflächen ein Gleitzustand eintreten kann [70, 73]. Dabei ist die Dynamik im Gleitzustand ein System der Ordnung $n - m$, was anhand des Systems zweiter Ordnung aus Abbildung 3.2 nachvollzogen werden kann. Die zu bestimmende Dynamik im Gleitzustand ist also von einer geringeren Ordnung als das zu regelnde System; außerdem werden für die Vorgabe der Gleitzustandsdynamik meist lineare Systeme verwendet. Beides führt besonders bei Systemen hoher Ordnung zu einer deutlichen Vereinfachung der Reglerauslegung [74]. Betrachtet man eine einzelne Schaltfunktion, respektive ein Eingrößensystem, so kann man diese als Linearkombination der Zustandsgrößen definieren:

$$S = \sum_{i=1}^n \lambda_i \cdot x^{(i-1)} = 0 \quad \text{mit} \quad \lambda_n = 1. \quad (3.8)$$

Demnach reduziert sich beispielsweise die Dynamik eines nichtlinearen Eingrößensystems zweiter Ordnung im Gleitzustand auf ein lineares System erster Ordnung, wobei die Koeffizienten λ_i dem Hurwitz-Kriterium genügen. Das Stabilitätskriterium nach Hurwitz seinerseits besagt, dass ein lineares System n -ter Ordnung dann asymptotisch stabil ist, wenn die Koeffizienten des charakteristischen Polynoms größer Null ($a_i > 0, i = 0, 1, \dots, n$) und die n führenden Hauptabschnittsdeterminanten D_i der Hurwitz-Matrix \mathbf{H} ebenfalls größer Null sind ($D_i > 0, i = 1, 2, \dots, n$) [28]. Außerdem ist ersichtlich, dass die Gleitzustandsdynamik durch die Parameter λ_i bestimmt wird und demnach bei der Auslegung vorgegeben werden kann.

Ein unerwünschter Effekt, der bei einem System im Gleitzustand auftreten kann, ist das sogenannte Rattern (engl. *chattering*). Dabei handelt es sich um eine hochfrequente Oszillation der Systemzustände um die Gleitlinie. Das Rattern tritt auf, wenn die Zustandsgrößen durch ständige Störungen oder ungenaue Modellierung der Systemdynamik immer wieder die Gleitlinie verlassen und vom diskontinuierlichen Stellgesetz wieder zurück auf die Gleitlinie gezwungen werden. Dieser Effekt kann durch eine nichtideale Eintreffphase in den Gleitzustand verstärkt werden, wenn die Zustände über die Gleitlinie hinaus streben. Weiterhin geht der ideale Gleitzustand von einer Umschaltung des Stellgesetzes mit unendlicher Schaltfrequenz aus, was in der realen Implementierung nicht verwirklicht werden kann. Die dennoch hohen Umschaltfrequenzen führen verständlicherweise zu erheblichen Belastungen und damit zum Verschleiß der Aktuatorik, weshalb das Rattern möglichst vermieden werden sollte. Es existieren verschiedene Ansätze, auftretendem Rattern entgegenzuwirken, etwa veränderte Stellgesetze oder eine direkte Vorgabe der Systemdynamik in der Eintreffphase des Gleitzustandes [70, 73–75]. Das Problem des Ratterns ist bei der Anwendung auf Lernverfahren neuronaler Netze von geringerer Bedeutung, da es keine zu verschleißende Aktuatorik gibt und man dem idealen Schaltverhalten eben dadurch deutlich näher kommt.

3.3.1 Existenz und Erreichbarkeit des Gleitzustandes

Bevor man ein Stellgesetz für eine definierte Schaltfunktion auslegen kann, muss zunächst nachgewiesen werden, dass der Gleitzustand existiert und dieser auch von allen Punkten des

Zustandsraumes erreicht werden kann. Zusätzlich muss gewährleistet sein, dass ein erreichter Gleitzustand nicht wieder verlassen wird. Kann beides gewährleistet werden, ist das im Gleitzustand geregelte System gemäß Abschnitt 3.2 global asymptotisch stabil [70–75]. Es ist leicht nachvollziehbar, dass diese Bedingung auch für ein neuronales Netz in einem Regelkreis von großer Bedeutung ist, da dadurch eine klare Stabilitätsaussage über das Lernverfahren gemacht werden kann. Zwei Ansätze zur Analyse von Existenz und Erreichbarkeit des Gleitzustandes werden häufig herangezogen [70, 71, 73, 74].

Direkter Ansatz über die Schaltfunktion:

Dieser Ansatz wird für jede der m Schaltfunktionen eines Systems durchgeführt; Existenz und Erreichbarkeit sind gegeben, wenn folgender Zusammenhang erfüllt ist:

$$S_i \cdot \dot{S}_i < 0 \quad \forall \quad i = 1, \dots, m. \quad (3.9)$$

Dies bedeutet, dass die Schaltfunktion und ihre Ableitung verschiedene Vorzeichen aufweisen:

$$\dot{S}_i < 0 \quad \text{wenn} \quad S_i > 0 \quad \text{und} \quad \dot{S}_i > 0 \quad \text{wenn} \quad S_i < 0. \quad (3.10)$$

Ljapunow-Ansatz:

Bei diesem Ansatz definiert man eine Ljapunow-Funktion $V(\vec{S})$, die die Bedingungen für global asymptotische Stabilität gemäß Abschnitt 3.2 erfüllt. Nach [25, 70, 71, 73, 75] lässt sich hierfür eine Funktion der folgenden Form verwenden:

$$V(\vec{S}) = \frac{1}{2} \cdot \vec{S}^T \cdot \vec{S}. \quad (3.11)$$

Zum Nachweis der Stabilität muss die Zeitableitung der Ljapunow-Funktion aus Gleichung 3.11 folglich kleiner Null sein.

$$\dot{V}(\vec{S}) = \dot{\vec{S}}^T \cdot \vec{S} < 0 \quad (3.12)$$

Aus diesem Zusammenhang wird deutlich, dass die beiden vorgestellten Ansätze für den Fall, dass es nur eine Schaltfunktion gibt ($m = 1$), identisch sind. Darüber hinaus gibt es einen Ansatz, der als *reaching law* bezeichnet wird [71, 73, 75] und zuweilen als eigenständige Methode zur Analyse der Existenz und Erreichbarkeit bezeichnet wird. Tatsächlich ist das *reaching law* jedoch eine Erweiterung des direkten Ansatzes über die Schaltfunktion, allerdings mit dem Vorteil, dass die Phase der Bewegung hin zum Gleitzustand näher beschrieben werden kann:

$$\dot{\vec{S}} = -\mathbf{Q} \cdot \text{sign}(\vec{S}) - \mathbf{K} \cdot \vec{h}(\vec{S}), \quad \text{mit} \quad (3.13)$$

$$\begin{aligned} \mathbf{Q} &= \text{diag}[q_1, \dots, q_m] \quad \text{und} \quad q_i > 0, \\ \text{sign}(\vec{S}) &= [\text{sign}(S_1), \dots, \text{sign}(S_m)], \\ \mathbf{K} &= \text{diag}[k_1, \dots, k_m] \quad \text{und} \quad k_i > 0, \\ \vec{h}(\vec{S}) &= [h_1(S_1), \dots, h_m(S_m)]. \end{aligned} \quad (3.14)$$

Durch diese Definition von $\dot{\vec{S}}$ ist es mit Hilfe der Entwurfparameter \mathbf{Q} , \mathbf{K} und $\vec{h}(\vec{S})$ möglich, die Dynamik, mit der sich das System auf den Gleitzustand zubewegt, und die der Eintreffphase vorzugeben [71, 73, 75]. Unter Inkaufnahme zusätzlicher Entwurfparameter kann so das Erreichen des Gleitzustandes beschleunigt und durch die verbesserte Vorgabe der Dynamik während der Eintreffphase dem Effekt des Ratterns vorgebeugt werden.

3.3.2 Invarianz und Robustheit

Allgemein kann man feststellen, dass ein System im Gleitzustand ein hohes Maß an Robustheit aufweist, da die Zustandstrajektorien nach einer Störung immer wieder zurück zur Gleitlinie streben. Diese Eigenschaft stellt einen der Hauptvorteile der Gleitzustandsregelung dar [69–71, 73–75]. Nach [71] ist die Gleitzustandsregelung robust gegenüber Parameterunsicherheiten, Nichtlinearitäten und unmodellierter Systemdynamik. In [73] und [75] wird darüber hinaus gezeigt, dass ein System im idealen Gleitzustand tatsächlich invariant, also vollkommen unempfindlich, gegenüber inneren und äußeren Störungen ist. Auch wenn der ideale Gleitzustand nicht gänzlich realisiert werden kann, so unterstreicht diese Aussage doch die Vorteile der Gleitzustandsregelung. Zur Betrachtung der Invarianz wird das System aus Gleichung 3.1 um Störeinflüsse erweitert, was folgende Zustandsgleichung liefert:

$$\dot{\vec{x}} = \vec{f}(\vec{x}, t) + \Delta\vec{f}(\vec{x}, \vec{p}, t) + [\mathbf{B}(\vec{x}, t) + \Delta\mathbf{B}(\vec{x}, \vec{p}, t)] \cdot \vec{u} + \vec{f}_s(\vec{x}, \vec{p}, t) . \quad (3.15)$$

Dabei repräsentiert \vec{p} vorhandene Störparameter, so dass man die Störungen mit $\Delta\vec{f}$ und $\Delta\mathbf{B}$ in innere sowie mit \vec{f}_s in äußere Störungen aufteilen kann. Man geht nun davon aus, dass jede Störung durch eine Kombination aus Eingangsmatrix \mathbf{B} und entsprechend wählbaren Koeffizienten darstellbar ist:

$$\begin{aligned} \Delta\vec{f}(\vec{x}, \vec{p}, t) &= \mathbf{B}(\vec{x}) \cdot \underline{\vec{f}}(\vec{x}, \vec{p}, t) , \\ \Delta\mathbf{B}(\vec{x}, \vec{p}, t) &= \mathbf{B}(\vec{x}) \cdot \underline{\mathbf{B}}(\vec{x}, \vec{p}, t) , \\ \vec{f}_s(\vec{x}, \vec{p}, t) &= \mathbf{B}(\vec{x}) \cdot \underline{\vec{f}}_s(\vec{x}, \vec{p}, t) . \end{aligned} \quad (3.16)$$

Diese Beschreibungen der Störungen können nun in Gleichung 3.15 eingesetzt werden, wobei die Klammern zur deutlicheren Darstellung entfallen.

$$\dot{\vec{x}} = \vec{f} + \mathbf{B} \cdot \underline{\vec{f}} + [\mathbf{B} + \mathbf{B} \cdot \underline{\mathbf{B}}] \cdot \vec{u} + \mathbf{B} \cdot \underline{\vec{f}}_s \quad (3.17)$$

Um die Invarianz zeigen zu können, wird auf das Konzept der äquivalenten Regelung aus Abschnitt 3.1 zurückgegriffen, womit die Dynamik im Gleitzustand beschrieben werden kann. Dazu wird Gleichung 3.3 herangezogen und obiger Zusammenhang eingesetzt.

$$\dot{\vec{S}}(\vec{x}) = \frac{\partial \vec{S}}{\partial \vec{x}} \cdot \left[\vec{f} + \mathbf{B} \cdot \underline{\vec{f}} + [\mathbf{B} + \mathbf{B} \cdot \underline{\mathbf{B}}] \cdot \vec{u} + \mathbf{B} \cdot \underline{\vec{f}}_s \right] \quad (3.18)$$

Stellt man diesen Ausdruck nun nach der Stellgröße \vec{u} um, so erhält man analog zu Gleichung 3.5 das äquivalente Stellgesetz unter Störeinfluss, was anschließend in Gleichung 3.17 eingesetzt werden kann. Nach Umstellen und Auflösen erhält man schließlich:

$$\dot{\vec{x}} = \vec{f}(\vec{x}, t) - \mathbf{B}(\vec{x}, t) \cdot \left(\frac{\partial \vec{S}}{\partial \vec{x}} \cdot \mathbf{B}(\vec{x}, t) \right)^{-1} \cdot \frac{\partial \vec{S}}{\partial \vec{x}} \cdot \vec{f}(\vec{x}, t) . \quad (3.19)$$

Betrachtet man diesen Ausdruck, ist sofort ersichtlich, dass er identisch mit Gleichung 3.6 ist, die Störungen also entfallen. Daran wird deutlich, dass ein System im Gleitzustand tatsächlich invariant gegenüber inneren und äußeren Störungen ist. An dieser Stelle ist anzumerken, dass dieses starke Robustheitsmaß nur für den Gleitzustand und nicht für die Zustandstrajektorien außerhalb desselben gilt. Daher sollte die Phase außerhalb des Gleitzustandes so kurz wie möglich gehalten werden. Eine Möglichkeit hierfür ist zum Beispiel die geeignete Wahl der Parameter λ_i , da diese Einfluss auf die zeitliche Änderung der Schaltfunktion \dot{S} besitzen.

3.4 Anwendung der Gleitzustandsregelung auf neuronale Netzwerke

Das Ziel dieses Kapitels ist es, das Konzept der Gleitzustandsregelung auf Lernverfahren neuronaler Netze zu übertragen. Um dies umsetzen zu können, ist ein Paradigmenwechsel hinsichtlich der Betrachtung von Trainingsverfahren neuronaler Netzwerke nötig. In Kapitel 2 ist dargestellt worden, dass es sich bei den gängigen Lernalgorithmen um mathematische Optimierungsverfahren handelt, die eine passende Gewichtungskonstellation finden sollen. Wie der Name bereits sagt, muss ein KNN unter Verwendung der strukturvariablen Regelung zur Gewichtseinstellung als geregeltes System betrachtet werden; es reicht demnach nicht länger aus, es nur als zu optimierendes System anzusehen. Damit stellen sich die gleichen Fragen nach der Charakterisierung der Systemdynamik und Beschaffenheit der Zustandsgrößen wie bei jedem anderen geregelten Prozess auch. Gleichzeitig verbessert aber die regelungstechnische Betrachtung von Lernverfahren das Verständnis über das Verhalten neuronaler Netzwerke in geschlossenen Regelkreisen. Dies geschieht zum Beispiel dadurch, dass die Lernrate nicht mehr empirisch ermittelt zu werden braucht, sondern über Stabilitätsforderungen bestimmbar ist. Der Vorwurf, ein KNN verhalte sich wie eine *black box*, ist wie sich zeigen wird unter diesen Voraussetzungen daher schwer aufrecht zu halten. Anzumerken ist außerdem, dass das Gleitzustandslernverfahren ein generischer Ansatz ist, der auf verschiedenste Trainingsprobleme neuronaler Netze angewendet werden kann [79–83]. Um den geregelten Prozess des Netzwerktrainings zu verdeutlichen, ist dieser in Abbildung 3.4 nun als allgemeines Blockschaltbild dargestellt.

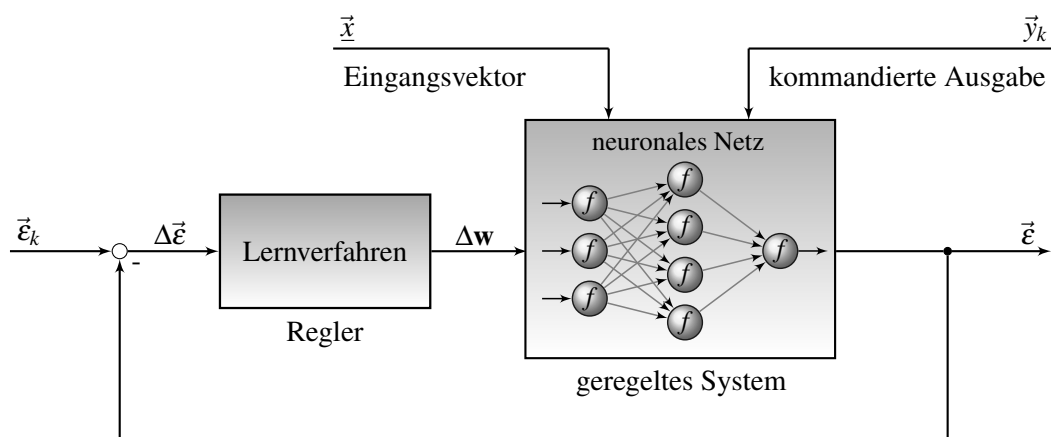


Abbildung 3.4: Das Training eines neuronalen Netzes als Regelungsprozess.

Man sieht, dass der Prozess des Netztrainings einen geschlossenen Regelkreis mit der üblichen Fehlerückführung darstellt. Dieser Regelkreis besteht aus dem geregelten System, also dem neuronalen Netzwerk und einem Regler, dem Gleitzustandslernverfahren. Als Ausgabe des geregelten Systems wird nun nicht die eigentliche Netzausgabe \vec{y} gesehen, sondern der Ausgabefehler \vec{e} . Dieser stellt folglich die Regelgröße des geschlossenen Kreises dar, wobei seine Dynamik durch den Eingangsvektor des KNN \vec{x} , die kommandierte Netzausgabe \vec{y}_k und die aktuelle Gewichtsconfiguration \mathbf{w} beeinflusst wird. Der Ausgabefehler wird durch die geschlossene Regelschleife mit der Führungsgröße \vec{e}_k verglichen, womit sich eine Regelabweichung $\Delta\vec{e}$ bestimmen lässt. Im Rahmen des Netzwerktrainings wird die Führungsgröße \vec{e}_k dabei in der Regel gleich Null sein. Die Aufgabe des Reglers besteht nun darin, die Stellgröße zu ermitteln, mit der eine existierende Regelabweichung minimiert wird. Dafür werden die Gewichtsänderungen $\Delta\mathbf{w}$ als Stellvektor aufgefasst, was insofern plausibel ist, als dass diese diejenigen Größen darstellen, die die Dynamik des Systems beeinflussen und nicht wie \vec{x} und \vec{y}_k von außen vorgegeben werden. Damit ist ein geschlossener Regelkreis definiert, auf den das Gleitzustandsverfahren angewendet werden kann.

Bevor die Vorteile des Gleitzustandes allerdings für neuronale Netzwerke genutzt werden können, ist die Frage zu beantworten, welche die eigentlichen Zustandsgrößen eines KNN sind, die in das Gleitregime gezwungen werden sollen. Aus Abbildung 3.2 ist ersichtlich, dass ein dynamisches System stabil ist, wenn sich seine Zustandsgrößen im Koordinatenursprung befinden, also gleich Null sind. Übertragen auf ein neuronales Netz lässt sich feststellen, dass dieses zweifellos stabil ist, wenn der Ausgabefehler \vec{e} gleich Null ist und in diesem Zustand verharrt, folglich auch $\dot{\vec{e}} = 0$ gilt. Die Dynamik und die Stabilität des Lernprozesses werden also durch den Ausgabefehler und sein zeitliches Verhalten beschrieben, weshalb es berechtigt ist, diesen und seine Ableitung als Zustandsgrößen des Netzwerkes zu betrachten. Gelingt es folglich, den Ausgabefehler in den Gleitzustand zu zwingen, verfügt das Lernverfahren selbst über eine intrinsische regelungstechnische Stabilitätsforderung, die global asymptotisch ist. Dadurch ist der Prozess der Gewichts Anpassung an ein klares Gütekriterium geknüpft, das eine transparentere Analyse von neuronalen Netzwerken in adaptiven Regelungssystemen ermöglicht. Im weiteren Verlauf wird sich darüber hinaus zeigen, dass die Vorteile dieses Lernverfahrens besonders in Kombination mit der dynamischen Inversion und der Stabilität ihrer Fehlerdynamik zum Tragen kommen.

3.4.1 Definition der Schaltfunktion

Der erste Schritt bei der Auslegung eines Gleitzustandsreglers ist die Definition der notwendigen Schaltfunktionen; dies besitzt auch für neuronale Netzwerke Gültigkeit. Macht man sich die obige Sichtweise des Trainings als geregelten Prozess zu eigen, so kann man die Gewichtsänderungen als einzelne Stellgröße auffassen, womit die behandelten Netzwerke regelungstechnisch als Eingrößensysteme betrachtet werden können. Folglich ist die Definition einer einzelnen Schaltfunktion erforderlich, was in der spärlich verfügbaren Literatur zu diesem Thema ein gängiger Ansatz ist. Diesbezüglich sind in [25, 68] und [84] gute Übersichten

zu allgemeinen Aspekten der Kombination von künstlicher Intelligenz und strukturvariabler Regelung gegeben. Hinsichtlich der Definition der Schaltfunktionen sind die existierenden Ansätze allerdings keineswegs einheitlich, was vor allem in der bereits diskutierten Wahl der Zustandsgrößen des neuronalen Netzwerkes begründet ist: In [26, 67, 85] und [86] wird der quadratische Fehler E verwendet, [87] schlägt die Gradientenmatrix $\nabla E(\mathbf{w})$ vor, [88] die Gewichtsänderung $\Delta \mathbf{w}$ und [89] sowie [90] den Ausgabefehler $\vec{\epsilon}$.

Die Wahl des Ausgabefehlers $\vec{\epsilon}$ und nicht einer anderen Größe als Netzwerkzustand lässt sich neben den zuvor zur Fehlerdynamik ausgeführten Gründen zusätzlich durch die Theorie der Gleitzustandsregelung begründen. Die Größen $\nabla E(\mathbf{w})$ und $\Delta \mathbf{w}$ weisen eine Anzahl an Elementen entsprechend der Menge an Verbindungsgewichten auf. Auch wenn diese durchaus das Verhalten des Netzwerkes beschreiben, müsste man genau genommen jedes einzelne Element als Zustandsgröße auffassen. Bei konsistenter Befolgung der theoretischen Annahmen führte dies zu einer Schaltfunktion extrem hoher Ordnung und selbst bei den in [87] und [89] getroffenen Vereinfachungen wird die entwurfstechnisch sehr aufschlussreiche Betrachtung der Zustände im Phasenraum nahezu unmöglich. Der quadratische Fehler E andererseits stellt wie die zuvor diskutierten Größen eine aus dem Ausgabefehler $\vec{\epsilon}$ berechnete Größe dar, bei der allerdings die Vorzeicheninformation entfallen ist, weshalb auch diese Wahl aus regelungstechnischer Sicht ebenfalls nicht schlüssig erscheint. Der Ausgabefehler $\vec{\epsilon}$ ist somit das unmittelbare Maß zur Charakterisierung der Fehlerdynamik eines neuronalen Netzwerkes. Zusätzlich ist die Wahl von $\vec{\epsilon}$ und nicht E als Zustandsgröße in [91] vergleichend untersucht worden, wobei sich das $\vec{\epsilon}$ nutzende Lernverfahren als durchgehend überlegen erwiesen hat.

Aus der Theorie ist deutlich geworden, dass nichtlineare Systeme durch Anwendung der Gleitzustandsregelung als lineare Systeme reduzierter Ordnung betrachtet werden können. Übertragen auf ein neuronales Netzwerk stellt sich also Frage, welche Ordnung dieses nichtlineare System besitzt und demzufolge auch von welcher Ordnung die lineare Schaltfunktion sein soll. Dabei kann es im Sinne der Komplexität des Lernverfahrens und damit seiner regelungstechnischen Transparenz nicht das Ziel sein, die Dynamik eines KNN exakt nachzubilden, sondern vielmehr sein Verhalten mit einer angemessenen Modellbeschreibung wiederzugeben. Auf Grund seines Aufbaus und seiner Übertragungseigenschaften lässt sich mit Sicherheit feststellen, dass ein neuronales Netz schwingungsfähiges Verhalten aufweist, wodurch die Betrachtung als System erster Ordnung nicht sinnvoll erscheint. Fasst man ein KNN folglich vereinfacht als nichtlineares System zweiter Ordnung auf, ergäbe sich als Dynamik der Schaltfunktion ein lineares System erster Ordnung. Mit dieser Modellannahme kann zum einen das schwingungsfähige Verhalten eines KNN abgebildet werden. Zum anderen muss zur Bestimmung der Netzzustände der Ausgabefehler $\vec{\epsilon}$ nur ein Mal abgeleitet werden und es ist eine anschauliche Betrachtung der Fehlerdynamik des KNN in einem Phasendiagramm möglich. Unter Nutzung des Zusammenhanges aus Gleichung 3.8 ergibt sich für die Schaltlinie S somit:

$$S = \dot{\vec{\epsilon}} + \lambda \cdot \vec{\epsilon} . \quad (3.20)$$

Bei $S = 0$ befindet sich das System auf der Gleitlinie, wo der Netzwerkfehler $\vec{\epsilon}$ gegen Null gleitet, wenn der Faktor λ das Hurwitz-Kriterium erfüllt. Dies wird auch deutlich, wenn man Gleichung 3.20 nach dem Ausgabefehler $\vec{\epsilon}$ auflöst:

$$S = \dot{\vec{\epsilon}} + \lambda \cdot \vec{\epsilon} = 0 \quad \Rightarrow \quad \vec{\epsilon}(t) = \vec{\epsilon}(t_0) \cdot e^{-\lambda(t-t_0)} . \quad (3.21)$$

Darüber hinaus muss die zweite Zustandsgröße des neuronalen Netzes $\dot{\vec{\epsilon}}$ bestimmt werden. Unter Annahme einer hohen Taktrate des Regelkreises und demzufolge einer kleinen Schrittweite T_s lässt sich schreiben:

$$\dot{\vec{\epsilon}} \approx \frac{\vec{\epsilon}(t) - \vec{\epsilon}(t - T_s)}{T_s} . \quad (3.22)$$

Für die Schaltlinie gemäß Gleichung 3.20 müssen nun die Existenz und die Erreichbarkeit nachgewiesen werden. Dazu wird der Ljapunow-Ansatz gemäß Abschnitt 3.3.1 herangezogen und für ein zeitdiskretes System angewendet. In Anlehnung an [26, 74, 89] und [92] kann man für die Existenz und Erreichbarkeit schließlich fordern, dass:

$$|S_{t+1}| < |S_t| . \quad (3.23)$$

Dabei beschreibt der Index t den aktuellen Zeit- beziehungsweise Lernschritt:

$$S(t) = S_t \quad \text{und} \quad S(t + T_s) = S_{t+1} . \quad (3.24)$$

Damit der Gleitzustand letztlich gemäß Gleichung 3.23 erreicht werden kann, muss nun ein Stellgesetz gefunden werden, das dies in Abhängigkeit der Schaltlinie gewährleistet.

3.4.2 Herleitung des Stellgesetzes

Da es sich bei der Anwendung der Gleitzustandsregelung auf neuronale Lernverfahren um eine generische Methode handeln soll, die für jedes Lernproblem angewendet werden kann, erscheint es zweckmäßig, einen existierenden Ansatz der Gewichtsanzpassung wenn möglich zu erweitern. Der Rückpropagationsalgorithmus aus Abschnitt 2.3 als etabliertes Trainingsverfahren bietet sich hierfür an, da mit der Gradientenbestimmung ein direktes Maß für den Einfluss jedes der Gewichte auf den Ausgabefehler gegeben ist. Die regelungstechnische Unschärfe dieses Verfahrens beruht maßgeblich auf der empirisch zu bestimmenden Lernrate. Die Lernregel aus Gleichung 3.25 ermöglicht es, dieses Problem durch Kombination des Rückpropagationsalgorithmus aus Gleichung 2.25 mit der Schaltlinie S_t zu umgehen.

$$\Delta \mathbf{w} = \left(\frac{\partial \vec{y}(\mathbf{w}, \vec{x})}{\partial \mathbf{w}} \right)^T \cdot \mu \cdot \text{diag}[\text{sign}(S_t)] \cdot |\vec{\epsilon}| \quad (3.25)$$

Die Wahl dieser Lernregel und ihre Leistungsfähigkeit ist dabei unter anderem in [79] umfangreich untersucht worden. Damit die Zustandsgrößen des KNN den Gleitzustand tatsächlich erreichen, muss die Ungleichung 3.23 in jedem Zeitschritt t erfüllt sein. Dazu müssen die Schaltfunktionen S_t und S_{t+1} genauer quantifiziert werden, was unter Zuhilfenahme der Gleichungen

3.20 und 3.22 folgende Zwischenlösung ergibt:

$$S_t = \dot{\tilde{\epsilon}}_t + \lambda \cdot \tilde{\epsilon}_t = \left(\lambda + \frac{1}{T_s} \right) \tilde{\epsilon}_t - \frac{1}{T_s} \tilde{\epsilon}_{t-1} , \quad (3.26)$$

$$S_{t+1} = \dot{\tilde{\epsilon}}_{t+1} + \lambda \cdot \tilde{\epsilon}_{t+1} = \left(\lambda + \frac{1}{T_s} \right) \tilde{\epsilon}_{t+1} - \frac{1}{T_s} \tilde{\epsilon}_t . \quad (3.27)$$

Der unbekannte Fehler $\tilde{\epsilon}_{t+1}$ aus Gleichung 3.27 muss nun approximiert werden, wofür folgender Ansatz gewählt wird:

$$\tilde{\epsilon}_{t+1} = \tilde{\epsilon}_t + \Delta \tilde{\epsilon}_t . \quad (3.28)$$

Die darin auftretende Änderung des Ausgabefehlers innerhalb eines Zeitschrittes $\Delta \tilde{\epsilon}_t$ kann nach Gleichung 3.29 berechnet werden.

$$\Delta \tilde{\epsilon}_t = \tilde{\epsilon}_{t+1} - \tilde{\epsilon}_t = (\vec{y}_{k,t+1} - \vec{y}_{t+1}) - (\vec{y}_{k,t} - \vec{y}_t) = \Delta \vec{y}_{k,t} - \Delta \vec{y}_t \quad (3.29)$$

Hierbei bezeichnet $\Delta \vec{y}_{k,t}$ die Änderung der kommandierten Netzwerkausgabe und $\Delta \vec{y}_t$ die Änderung der tatsächlichen Netzwerkausgabe innerhalb eines Zeitschrittes T_s . Für die Bestimmung von $\Delta \vec{y}_t$ wird eine Taylorreihenentwicklung durchgeführt:

$$\Delta \vec{y}_t = \frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \mathbf{w}_t} \cdot \Delta \mathbf{w}_t + \frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t . \quad (3.30)$$

Die partiellen Ableitungen $\frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \mathbf{w}_t}$ und $\frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \vec{x}_t}$ werden bei der Bestimmung der aktuellen Gewichtsänderung $\Delta \mathbf{w}_t$ gemäß der in Abschnitt 2.3.1 beschriebenen Prozedur bestimmt, so dass diese drei Größen vorliegen. Die Änderung der Netzwerkeingänge $\Delta \vec{x}_t$ kann entweder aus der Dynamik des übergeordneten geregelten Systems abgeleitet oder für kleine Zeitschritte T_s als minimal angenommen werden. Die Änderung der kommandierten Netzwerkausgabe $\Delta \vec{y}_{k,t}$ kann in Abhängigkeit der Fehlerdynamik des KNN und der Dynamik des übergeordneten Systems beliebig vorgegeben werden. Eine einfache, aber effektive Vorgabe lässt sich bereits mit der Gradienteninformation des zurückliegenden Zeitschrittes erreichen, indem diese einen Zeitschritt in die Zukunft propagiert wird. Zur weiteren Berechnung der Erreichbarkeit des Gleitzustandes werden nun Gleichungen 3.27 und 3.28 in die Ungleichung 3.23 eingesetzt:

$$|S_t| > \left| \left(\lambda + \frac{1}{T_s} \right) (\tilde{\epsilon}_t + \Delta \tilde{\epsilon}_t) - \frac{1}{T_s} \tilde{\epsilon}_t \right| . \quad (3.31)$$

Hier können jetzt weiterhin die hergeleiteten Zusammenhänge der Gleichungen 3.29 und 3.30 eingesetzt werden:

$$|S_t| > \left| \left(\lambda + \frac{1}{T_s} \right) \left(\tilde{\epsilon}_t + \Delta \vec{y}_{k,t} - \frac{\partial \vec{y}_t}{\partial \mathbf{w}_t} \cdot \Delta \mathbf{w}_t - \frac{\partial \vec{y}_t}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right) - \frac{1}{T_s} \tilde{\epsilon}_t \right| . \quad (3.32)$$

In obigen Ausdruck lässt sich unter Anwendung der Substitution $\left(\frac{\partial \vec{y}_t(\mathbf{w}_t, \vec{x}_t)}{\partial \mathbf{w}_t} \right)^T = \nabla \vec{y}_t$ nun die Lernregel aus Gleichung 3.25 einsetzen.

$$|S_t| > \left| \left(\lambda + \frac{1}{T_s} \right) \left(\tilde{\epsilon}_t + \Delta \vec{y}_{k,t} - \frac{\partial \vec{y}_t}{\partial \mathbf{w}_t} \cdot \left[\nabla \vec{y}_t \cdot \mu \cdot \text{diag}[\text{sign}(S_t)] \cdot |\tilde{\epsilon}_t| \right] - \frac{\partial \vec{y}_t}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right) - \frac{1}{T_s} \tilde{\epsilon}_t \right| \quad (3.33)$$

Anhand dieser Ungleichung ist erkennbar, dass der einzige unbekannte Parameter, der bestimmt werden muss, damit die Erreichbarkeit des Gleitzustandes gewährleistet werden kann, die Lernrate μ ist. Die Herleitung des diskontinuierlichen Stellgesetzes des strukturvariablen Lernverfahrens mit Gleitzustand bedingt folglich die Bestimmung der Lernrate unter der Randbedingung global asymptotischer Stabilität. Bei der Umstellung der Ungleichung und der Berechnung der Lernrate 3.33 kommt außerdem der Aspekt vereinfachend hinzu, dass nur Netzwerke mit einem Neuron in der Ausgangsschicht ($i = 1$) verwendet werden. Zur übersichtlicheren Darstellung werden im Folgenden die Parameter a_i und b_i eingeführt:

$$a_i = \left(\lambda + \frac{1}{T_s} \right) \cdot \left(\varepsilon_{t,i} + \Delta y_{k,t,i} - \left[\frac{\partial \vec{y}_t}{\partial \vec{x}_t} \cdot \Delta \vec{x}_t \right]_i \right) - \frac{1}{T_s} \varepsilon_{t,i} , \quad (3.34)$$

$$b_i = \left(\lambda + \frac{1}{T_s} \right) \cdot \left(\frac{\partial \vec{y}_t}{\partial \mathbf{w}_t} \left(\frac{\partial \vec{y}_t}{\partial \mathbf{w}_t} \right)^T \cdot \text{diag} [\text{sign}(S_t)] \cdot |\vec{\varepsilon}_t| \right)_i . \quad (3.35)$$

Diese Substitutionen können in Gleichung 3.33 eingesetzt werden, womit man folgende vereinfachte Ungleichung zur Bestimmung der Lernrate μ erhält:

$$|S_{t,i}| > |a_i - \mu \cdot b_i| . \quad (3.36)$$

Für die Ungleichung 3.36 sind nun zwei Fälle zu unterscheiden:

$$\left\{ \begin{array}{ll} \textbf{Fall 1:} & S_{t,i} > 0 \\ \textbf{Fall 2:} & S_{t,i} < 0 \end{array} \right. . \quad (3.37)$$

Fall 1: Wendet man die Bedingung $S_{t,i} > 0$ auf Gleichung 3.36 an, folgt daraus:

$$S_{t,i} > |a_i - \mu \cdot b_i| = S_{t+1,i} . \quad (3.38)$$

Daraus lassen sich erneut zwei Fälle ableiten, nämlich:

$$\textbf{a: } S_{t+1} > 0 \quad \text{und} \quad \textbf{b: } S_{t+1} < 0 . \quad (3.39)$$

In der obigen Gleichung ist der Index i im Sinne der Deutlichkeit ausgelassen worden. Die Fälle a und b werden nun getrennt untersucht.

Fall 1a: Aus den Forderungen $S_{t,i} > 0$ und $S_{t+1} > 0$ ergibt sich unter der Bedingung $|S_{t+1}| < |S_t|$ aus Gleichung 3.23, dass

$$S_{t,i} > a_i - \mu \cdot b_i , \quad (3.40)$$

wobei aus $S_{t+1} > 0$ gleichzeitig der Zusammenhang gemäß Gleichung 3.41 folgt.

$$a_i - \mu \cdot b_i > 0 \quad (3.41)$$

Fall 1b: Analog zu obigem Fall folgt aus $S_{t,i} > 0$ und $S_{t+1} < 0$ sowie Gleichung 3.23:

$$S_{t,i} > -a_i + \mu \cdot b_i, \quad (3.42)$$

während aus $S_{t+1} < 0$ folgt, dass

$$a_i - \mu \cdot b_i < 0. \quad (3.43)$$

Fasst man die Ergebnisse von Fall 1 zusammen, können in Abhängigkeit von b_i Bereiche für die Lernrate gemäß Gleichung 3.44 festgelegt werden.

$$\left\{ \begin{array}{l} \left\{ \mu > -\frac{S_{t,i}-a_i}{b_i} \wedge \mu < \frac{a_i}{b_i} \right\} \vee \left\{ \mu < \frac{S_{t,i}+a_i}{b_i} \wedge \mu > \frac{a_i}{b_i} \right\} \quad \text{für } b_i > 0 \\ \left\{ \mu < -\frac{S_{t,i}-a_i}{b_i} \wedge \mu > \frac{a_i}{b_i} \right\} \vee \left\{ \mu > \frac{S_{t,i}+a_i}{b_i} \wedge \mu < \frac{a_i}{b_i} \right\} \quad \text{für } b_i < 0 \end{array} \right. \quad (3.44)$$

Daraus ist ersichtlich, dass die Terme $-\frac{S_{t,i}-a_i}{b_i} = -\frac{S_{t,i}}{b_i} + \frac{a_i}{b_i}$ und $\frac{S_{t,i}+a_i}{b_i} = \frac{S_{t,i}}{b_i} + \frac{a_i}{b_i}$ die eigentlichen Grenzen darstellen, so dass folgende Vereinfachung der Fallunterscheidung möglich ist:

$$\left\{ \begin{array}{l} -\frac{S_{t,i}-a_i}{b_i} < \mu < \frac{S_{t,i}+a_i}{b_i} \quad \text{für } b_i > 0 \\ \frac{S_{t,i}+a_i}{b_i} < \mu < -\frac{S_{t,i}-a_i}{b_i} \quad \text{für } b_i < 0 \end{array} \right. \quad (3.45)$$

Fall 2: Die Bedingung $S_{t,i} < 0$ wird auf Gleichung 3.36 angewendet, woraus folgt, dass

$$-S_{t,i} > |a_i + \mu \cdot b_i|. \quad (3.46)$$

Analog zu Fall 1 ergeben sich daraus zwei weitere Fallunterscheidungen.

$$S_{t+1} > 0 \quad \text{und} \quad S_{t+1} < 0 \quad (3.47)$$

Fall 2a: Aus den Forderungen $S_{t,i} < 0$ und $S_{t+1} > 0$ sowie der Stabilitätsbedingung aus Gleichung 3.36 folgt:

$$-S_{t,i} > a_i - \mu \cdot b_i. \quad (3.48)$$

Außerdem bedingt $S_{t+1} > 0$, dass

$$a_i - \mu \cdot b_i > 0. \quad (3.49)$$

Fall 2b: Zuletzt folgt aus $S_{t,i} < 0$ und $S_{t+1} < 0$ und der Stabilitätsbedingung folgender Bereich für die Lernrate:

$$-S_{t,i} > -a_i + \mu \cdot b_i. \quad (3.50)$$

Für $S_{t+1} < 0$ folgt schließlich:

$$a_i - \mu \cdot b_i < 0. \quad (3.51)$$

Diese Fälle können nun in Gleichung 3.52 zusammengefasst werden.

$$\left\{ \begin{array}{ll} \left\{ \mu > \frac{S_{t,i}+a_i}{b_i} \wedge \mu < \frac{a_i}{b_i} \right\} \vee \left\{ \mu < \frac{-S_{t,i}+a_i}{b_i} \wedge \mu > \frac{a_i}{b_i} \right\} & \text{für } b_i > 0 \\ \left\{ \mu < \frac{S_{t,i}+a_i}{b_i} \wedge \mu > \frac{a_i}{b_i} \right\} \vee \left\{ \mu > \frac{-S_{t,i}+a_i}{b_i} \wedge \mu < \frac{a_i}{b_i} \right\} & \text{für } b_i < 0 \end{array} \right. \quad (3.52)$$

Wie beim Fall 1 können die Bereiche der stabilen Lernrate vereinfacht betrachtet werden.

$$\left\{ \begin{array}{ll} \frac{S_{t,i}+a_i}{b_i} < \mu < \frac{-S_{t,i}+a_i}{b_i} & \text{für } b_i > 0 \\ -\frac{S_{t,i}+a_i}{b_i} < \mu < \frac{S_{t,i}+a_i}{b_i} & \text{für } b_i < 0 \end{array} \right. \quad (3.53)$$

Fasst man nun noch die Gleichungen 3.45 und 3.53 zusammen, kann man die Fälle 1 und 2 gemeinsam darstellen.

$$\left\{ \begin{array}{l} -\frac{S_{t,i}}{b_i} + \frac{a_i}{b_i} < \mu < \frac{S_{t,i}}{b_i} + \frac{a_i}{b_i} \text{ für } (S_{t,i} > 0 \wedge b_i > 0) \vee (S_{t,i} < 0 \wedge b_i < 0) \\ \frac{S_{t,i}}{b_i} + \frac{a_i}{b_i} < \mu < -\frac{S_{t,i}}{b_i} + \frac{a_i}{b_i} \text{ für } (S_{t,i} > 0 \wedge b_i < 0) \vee (S_{t,i} < 0 \wedge b_i > 0) \end{array} \right. \quad (3.54)$$

Damit sind klare Wertebereiche hergeleitet worden, in denen sich je nach auftretendem Fall die Lernrate befinden muss, damit die Erreichbarkeit des Gleitzustandes gewährleistet ist. Neben dem Vorteil der Stabilität sorgt dieser automatische Prozess auch dafür, dass die in Abschnitt 2.3 erwähnten Probleme zu kleiner oder zu großer Lernrate umgangen werden und die Konvergenzgeschwindigkeit des Algorithmus' daher in der Regel höher ist. Dies liegt auch darin begründet, dass eine aufwändige empirische und sich von System zu System unterscheidende Bestimmung der Lernrate durch einen automatischen, stabilitätsgebundenen Prozess ersetzt wird. Die Einsatzmöglichkeit von neuronalen Netzen, die das Gleitzustandslernverfahren nutzen, ist folglich besonders für die Regelung flexibler, da keine Optimierung der Lernrate bezüglich der Dynamik des zu regelnden Systems durchgeführt zu werden braucht.

3.5 Darstellung neuronaler Netzwerke im Gleitzustand

Im Folgenden soll exemplarisch gezeigt werden, dass die Zustände eines neuronalen Netzes mit dem vorgestellten Lernverfahren in den Gleitzustand gebracht werden und dass dieser Ansatz unter dem Einfluss externer Störungen dem Rückpropagationsalgorithmus überlegen ist. Da die Netzwerke im Regelkreis nichtlineare Zusammenhänge eines verkoppelten Mehrgrößensystems abbilden sollen, werden in allen Beispielen nichtlineare Zielfunktionen aus dynamischen Eingangssignalen abgebildet. Dabei lernen alle Netzwerke in sequentieller Form, wie es auch im geregelten System stattfindet. Zunächst werden zwei Netze mit 20 Neuronen in einer verdeckten Schicht verglichen, bei denen der Gleitzustand unter Verwendung unterschiedlicher Schaltfunktionen erreicht wird. Betrachtet man hierzu Gleichung 3.20, ist zu erkennen, dass die Dynamik der Schaltfunktion durch die Wahl des Parameters λ beeinflusst werden kann. Zur deutlichen Unterscheidung dieses Einflusses wird mit $\lambda_1 = 10$ ein relativ geringer und mit $\lambda_2 = 100$ ein hoher Wert gewählt. In Abbildung 3.5 sind nun die Zielfunktion und die beiden

Netzantworten sowie die dazugehörigen Eingangssignale für den zehnssekündigen Testfall dargestellt. Die Darstellung der Eingangssignale erfolgt zweigeteilt, für den unverrauschten und den verrauschten Fall, welcher im weiteren Verlauf für die Auswirkung externer Störungen verwendet wird. Die Zielfunktion setzt sich aus einer sinusförmigen Schwingung zusammen,

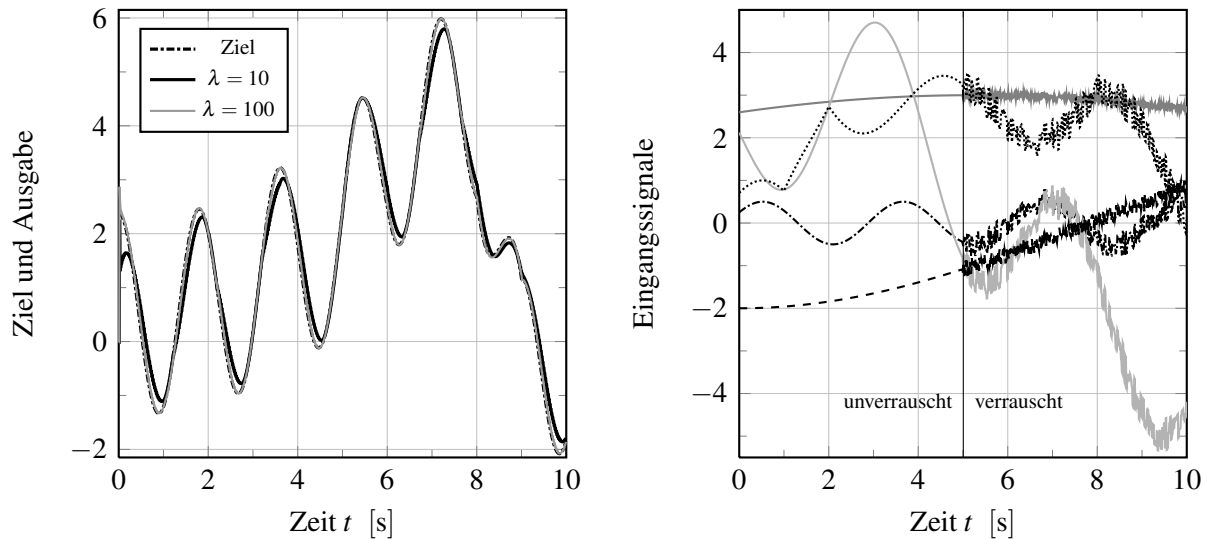


Abbildung 3.5: Zielfunktion und Antworten bei unterschiedlichem λ sowie Eingangssignale.

die mit rampenförmigen Signalen überlagert ist, wodurch die erkennbaren Verzerrungen entstehen. Bei den Eingangssignalen handelt es sich ebenfalls um Schwingungen unterschiedlicher Frequenz, Amplitude und Phasenlage, denen ebenfalls zusätzliche Signale beigemischt werden und die zunächst unverrauscht genutzt werden. Dadurch ist ein in der Komplexität anwendungsnaher Trainingsfall gegeben, der gleichzeitig eine transparente Analyse des Lernvorganges ermöglicht. Der Lernvorgang erfolgt nach zufälliger Initialisierung der Gewichte mit 100 Hz, so dass 1000 Gewichtsänderungen im Betrachtungszeitraum erfolgen. An den Netzantworten aus Abbildung 3.5 sind deutliche Unterschiede des Lernerfolges bei verschiedener Dynamik der Schaltfunktion zu erkennen. Während bei einem hohem Wert für den Parameter λ ein schnelles und direktes Abbilden der Zielfunktion erfolgt, kann die Netzausgabe bei kleinerem λ den Zielwerten besonders bei Richtungsänderungen der Kurve nur zeitverzögert folgen.

Der Einfluss von λ auf das Lernverhalten wird auch deutlich, wenn man das Phasendiagramm der beiden Testfälle betrachtet, welches zusammen mit den dazugehörigen Lernraten in Abbildung 3.6 gegeben ist. Hier sind die Trajektorien der Zustandsgrößen der Netzwerke und die jeweiligen Schaltlinien dargestellt. Wegen der zufällig initialisierten Gewichte beginnen beide Verläufe bei einem sprungförmigen Fehler von ungefähr 2,5. Da es kein Vorwissen im Netzwerk hinsichtlich des dynamischen Verhaltens des Zielwertes gibt und im ersten Lernschritt zunächst S_t aus Gleichung 3.23 bestimmt wird, schießen beide Trajektorien über ihre Schaltlinie hinaus. Ab dem zweiten Lernschritt ist dann für beide Verläufe der Effekt der Stabilitätsforderung aus

Gleichung 3.23 in einer Reduktion von $|S|$ mit jedem Zeitschritt und einer Bewegung hin zur Schaltlinie zu beobachten. Deutlich ist ein Unterschied in der Dynamik der Zustände zu erkennen: Während die Trajektorie bei hohem λ nach dem Überschwinger des ersten Lernschrittes in kürzester Zeit entlang der Gleitlinie in Richtung Koordinatenursprung strebt, verzeichnet die zweite Trajektorie zunächst einen Wechsel des Vorzeichens der Schaltfunktion. Erst dann strebt sie zur Gleitlinie, um sich mit mehreren Schritten dem Ursprung zu nähern. Zusätzlich wird dabei die in Abbildung 3.5 bereits erkennbare Verzögerung beim Erreichen der Zielwerte anhand eines breiteren Korridors deutlich, in dem begrenztes Rattern auftritt. An den berechneten Lernraten beider Schaltfunktionen aus Abbildung 3.6 wird weiterhin deutlich, dass es zum Einhalten der Stabilitätsforderung auch zu einmaligen oder kurzzeitigen Sprüngen kommen kann, was in Ausnahmefällen zu Unterschieden von bis zu drei Größenordnungen führt. Außerdem ist anzumerken, dass sich die Verläufe der Lernrate je nach den Charakteristiken des gestellten Trainingsproblems häufig auch stark voneinander unterscheiden können.

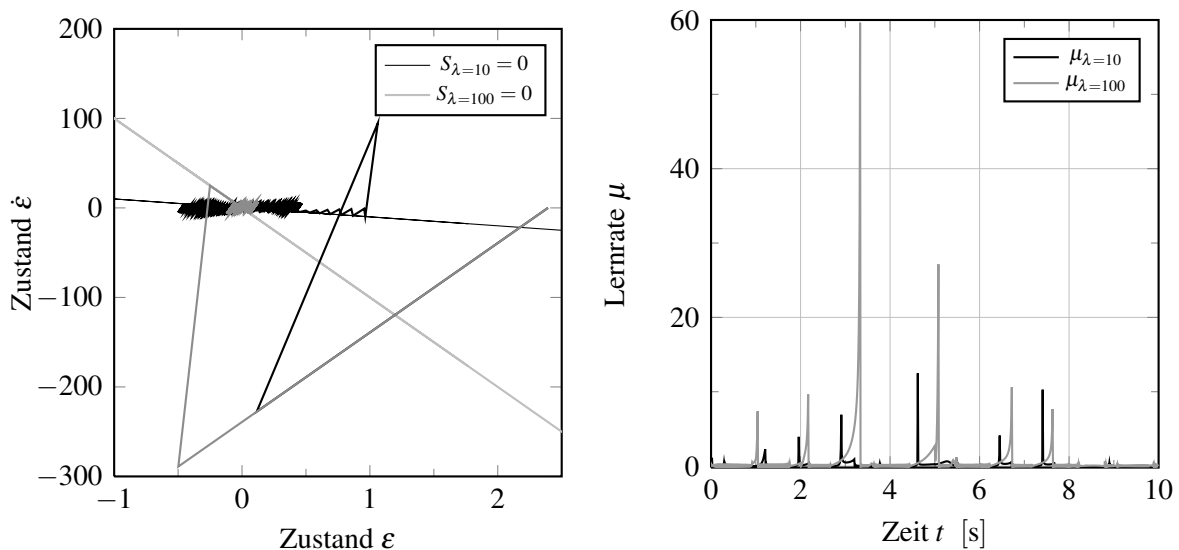


Abbildung 3.6: Phasendiagramm und Lernraten bei unterschiedlicher Schaltfunktion.

Durch das Gleitzustandslernverfahren werden die Zustände eines KNN folglich mit einer einstellbaren Dynamik in Richtung des stabilen Ursprunges im Phasenraum bewegt. Dabei ist eine sehr schnelle Dynamik durch eine hohe Wahl von λ allerdings nicht automatisch optimal, da dadurch auch Überschwinger beim Erreichen der Zielwerte begünstigt werden können. Beim folgenden Vergleich zwischen Gleitzustands- und Rückpropagationsverfahren in Gegenwart externer Störungen wird daher $\lambda = 40$ gewählt. In Abbildung 3.7 sind dazu die Netzantworten und die zugehörigen Phasendiagramme für einen Sprung in der Zielgröße bei vier Sekunden dargestellt. Da der Verlauf davor mit dem aus Abbildung 3.5 identisch ist, wird dieser im Sinne der Deutlichkeit ausgelassen. Durch den Sprung, der eine zusätzliche rampenförmige Änderung der Zielgrößen einleitet, wird eine erhebliche Störung eingebracht, die beide Lernverfahren zunächst sehr gut ausgleichen können. Erneut ist ein begrenztes Überschwingen des Gleitzustandsverfahrens ersichtlich, während der Ansatz konstanter Lernrate eher eine Zeitverzögerung

erkennen lässt. Bei diesem Beispiel tritt beim Rückpropagationsalgorithmus jedoch bei nahezu zehn Sekunden ein unkontrollierter Rauscheffekt auf; dieser kann bei vielen verschiedenen Lernproblemen, bei denen eine ungünstige Konstellation aus Eingangssignalen und Zielwerten mit zu hohen Lernraten kombiniert ist, reproduziert werden. Das Lernen mit Gleitzustand weist diese Störung der Netzausgabe auf Grund der stabilitätsbedingten variablen Lernrate nicht auf. Das Ausmaß der Störung wird auch bei Betrachtung des Phasendiagrammes deutlich: Während die Trajektorie für $\lambda = 40$ nach der Störung sofort wieder in den Gleitzustand übergeht und dauerhaft dort verbleibt, sind für die konstante Lernrate Ausschläge in der Phasebene ersichtlich, die nahezu den Wert der zu Grunde liegenden Störung einnehmen. Dass ein solches Ausgabeverhalten in einem geschlossenen Regelkreis zu Destabilisierungen führen kann und daher ausgeschlossen sein muss, ist leicht ersichtlich.

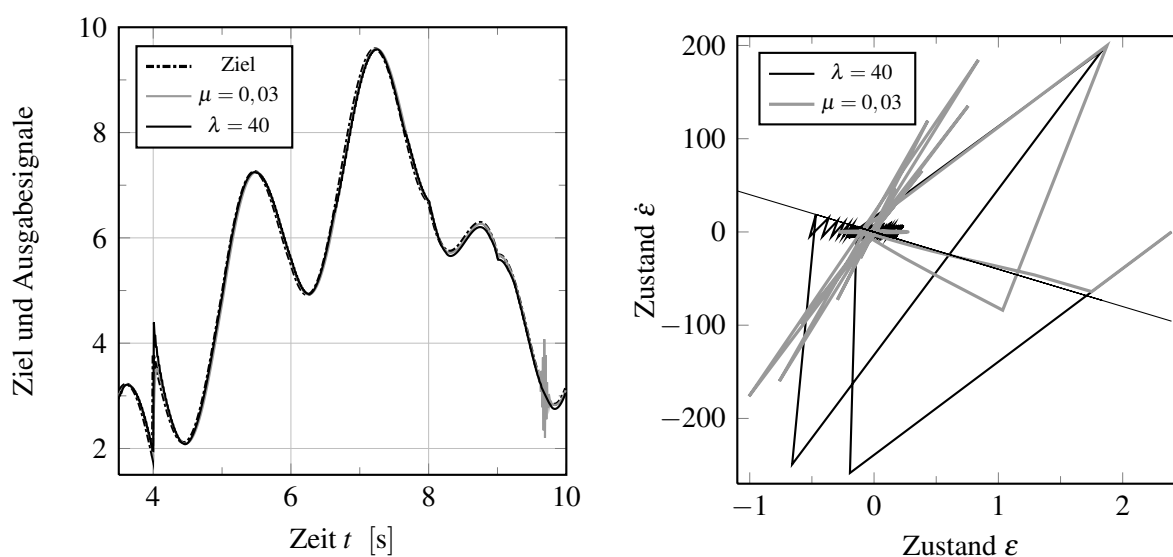


Abbildung 3.7: Netzantworten und Phasendiagramm bei einem Sprung der Zielgröße.

Um derartige Effekte dauerhaft ausschließen zu können, muss eine geringere Lernrate gewählt werden, welche zu große Sprünge in den Gewichtswerten vermeidet. Dadurch werden auch zu große Bewegungen in der Fehlerfläche ausgeschlossen, die bei ungünstiger Konstellation mit den Eingangsgrößen das beschriebene Rauschen begünstigen. Diese konservative Auslegung der Lernrate sorgt wiederum dafür, dass der Rückpropagationsalgorithmus in den meisten Fällen eine geringere Konvergenzgeschwindigkeit aufweist als das Lernverfahren auf Basis der Gleitzustandsregelung. Als Beispiele dafür sind in Tabelle 3.1 resultierende Trainingsfehler für das Lernproblem aus Abbildung 3.7 bei verschiedenen Rauschintensitäten der Eingangssignale gegeben. Die Rauschintensität von Zwei entspricht dabei der Darstellung aus Abbildung 3.5. Als Gütemaß werden sowohl der mittlere quadratische Fehler (engl. *mean square error*, MSE) als auch die Wurzel desselben (engl. *root mean square error*, RMSE) gewählt, wobei das Rückpropagationsverfahren mit RP und der Ansatz auf Basis der Gleitzustandsregelung mit GZR gekennzeichnet sind. Als Lernrate wurde mit $\mu = 0,275$ der höchste Wert gewählt, bei dem im Rahmen der empirischen Betrachtung kein Rauscheffekt im Ausgangssignal auftritt.

| Rauschintensität | MSE_{GZR} | $RMSE_{GZR}$ | MSE_{RP} | $RMSE_{RP}$ |
|------------------|-------------|--------------|------------|-------------|
| 0 | 0,035 | 0,188 | 0,050 | 0,255 |
| 1 | 0,040 | 0,206 | 0,057 | 0,239 |
| 1,5 | 0,045 | 0,212 | 0,061 | 0,247 |
| 2 | 0,050 | 0,225 | 0,063 | 0,252 |

Tabelle 3.1: Vergleich der Fehlerwerte bei verschiedenen Rauschintensitäten.

Man sieht, dass die Ergebnisse des RP-Verfahrens von geringerer Güte sind als diejenigen des GZR-Ansatzes. Der RMSE, der den Fehler in der gleichen physikalischen Dimension wie die der Zieldaten repräsentiert, ist im ungünstigsten Rauschfall beim RP-Verfahren etwa 10 % höher, bei weniger verrauschten Eingangssignalen steigt der Wert weiter auf bis zu 35 %. Für die Rauschintensität von Zwei sind in Abbildung 3.8 zusätzlich die Netzausgaben beider Verfahren dargestellt. Deutlich zu erkennen ist, wie sich bei beiden der Sprung in den Zielwerten mit einem begrenzten Rauschen auf die Netzausgabe auswirkt. Wie die Abbildung und die Ergebnisse aus Tabelle 3.1 zeigen, neigt das Gleitzustandslernverfahren allerdings deutlich weniger zu unerwünschten verrauschten Netzausgaben. Daran lässt sich erkennen, dass die in der Theorie beschriebene Robustheit eines Systems im Gleitzustand gegenüber externen Störungen auch für neuronale Netzwerke gegeben ist. Zusätzlich sorgt die variable Lernrate für eine erhöhte Konvergenzgeschwindigkeit dieses Lernverfahrens. Es ist leicht ersichtlich, dass dies in einem Regelkreis besonders dann von großer Relevanz ist, wenn sprunghafte Störungen zum Beispiel durch Beschädigungen am zu regelnden System auftreten und dieses destabilisieren.

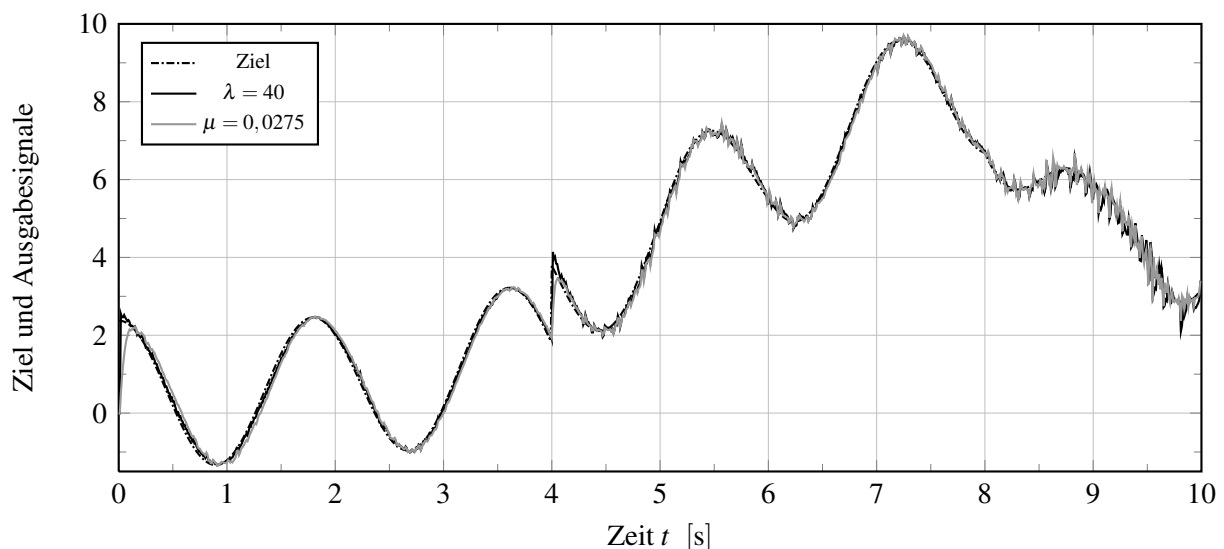


Abbildung 3.8: Netzantworten bei einem Sprung der Zielgröße und zusätzlichem Rauschen.

4 Flugsimulation und Reglerimplementierung

Die Bewegung eines Flugzeuges im Raum kann als System mit sechs Freiheitsgraden aufgefasst werden, dessen Zustandsänderungen durch zwölf nichtlineare Differentialgleichungen beschrieben sind. Bei diesem Gesamtsystem Flugzeug treten eine Reihe von Kopplungseffekten zwischen den einzelnen Bewegungsformen auf, während gleichzeitig schwingungsfähige Teilsysteme existieren wie zum Beispiel die Anstellwinkel- oder die Taumelschwingung [28, 93]. Betrachtet man das Flugzeug nun als Regelstrecke, so handelt es sich wegen dieser Eigenschaften um ein komplexes Mehrgrößensystem, für dessen Regelung vielfältige Modellannahmen getroffen werden müssen. Gleichzeitig erfordert die Auslegung von Flugreglern in der Regel ein hohes Maß an Systemkenntnis, welche unter anderem durch eine aufwändige Identifizierung der aerodynamischen Parameter des Flugzeuges gewonnen werden kann. Diese Gesichtspunkte sorgen dafür, dass der Entwurf eines Flugregelungssystems einen iterativen Prozess darstellt, bei stets auch Kompromisse zwischen gegensätzlichen Forderungen an die Regelgüte gemacht werden müssen.

Da die systematische Untersuchung von Reglereigenschaften folglich mit nicht unerheblichem Aufwand verbunden ist, ist es ingenieurtechnischer Standard, dies, wie bei anderen komplexen Regelungsaufgaben auch, mit Hilfe der nichtlinearen Simulation durchzuführen. Diese geht der Erprobung im Flugversuch stets voraus und sollte im Sinne aussagekräftiger Ergebnisse ein möglichst realitätsnahes Abbild der Wirklichkeit sein, also die für die Reglerbewertung maßgeblichen Effekte möglichst genau darstellen. Die in dieser Arbeit diskutierten Regelungsansätze sind daher in umfangreichen Flugsimulationen validiert worden, wobei der prinzipielle Aufbau der Simulationsumgebung in Abbildung 4.1 dargestellt ist. Es ist erkennbar, dass diese aus fünf Hauptblöcken besteht, nämlich der Flugzeugdynamik, der Modellierung der Sensorik, dem eigentlichen Flugregler, den Aktuatormodellen und einem Wind- und Turbulenzmodell zur Bestimmung adäquater Störgrößen. Einer der Kernbestandteile der Simulation ist der Block der Flugzeugdynamik, da in ihm auf Grundlage des Kräfte- und Momentengleichgewichts und der Kenntnis der aerodynamischen Eigenschaften des Flugzeuges die kinematischen Größen berechnet werden. Dabei ist der Startpunkt eines jeden Simulationslaufes eine Trimmrechnung, bei der die Trimmgrößen des aktuellen Arbeitspunktes aus den jeweiligen Trimmforderungen berechnet werden [28]. Darüber hinaus handelt es sich bei der dargestellten Flugsimulation um ein generisches Verfahren, das auf beliebige Flugzeugtypen angewendet werden kann. Da die Bewertung der Regelungsstrategien auf Basis dieser Simulation erfolgt, sollen ihre wichtigsten Bestandteile und zu Grunde liegenden Annahmen nun kurz skizziert werden.

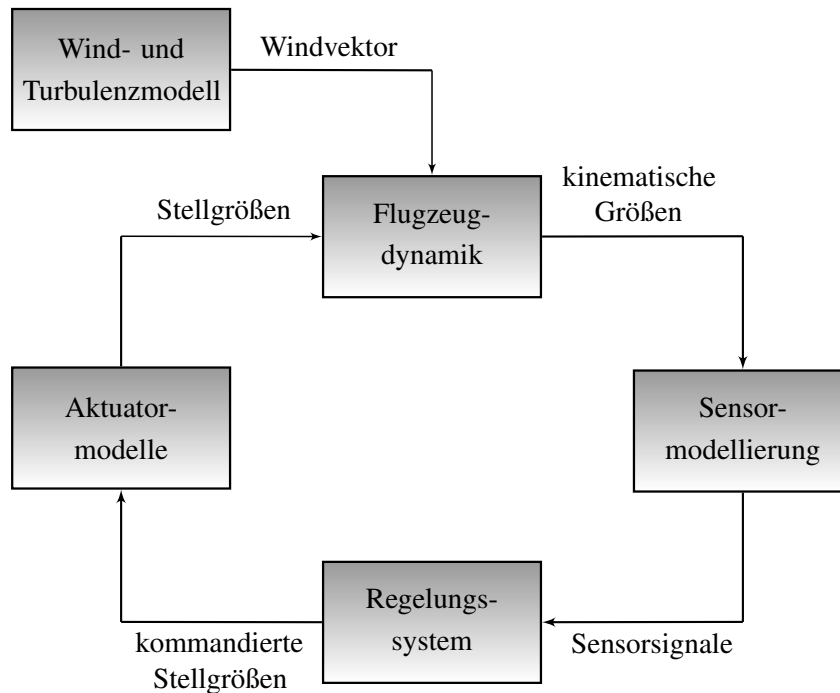


Abbildung 4.1: Übersichtsdarstellung der nichtlinearen Flugsimulation.

4.1 Nichtlineare Zustandsgleichungen der Flugzeugbewegung

Bevor die zwölf nichtlinearen Differentialgleichungen zur Beschreibung der Flugzeugbewegung dargestellt werden, sind zunächst einige wichtige vereinfachende Modellannahmen zu treffen, wie sie für die Flugregelungsanwendung üblich sind [28, 52, 54, 93]:

- Eine ruhende, glatte Erde wird als Inertialsystem verwendet, womit alle Effekte, die aus der Erdrotation herrühren, als vernachlässigbar angesehen werden.
- Das Flugzeug wird als starrer Körper mit konstanter Masse angesehen, wodurch aeroelastische Effekte nicht berücksichtigt werden.
- Die Erzeugung der aerodynamischen Kräfte in Folge der Anströmung wird als quasistationär angenommen.
- Alle am Flugzeug angreifenden Kräfte werden als im Schwerpunkt konzentriert angesehen.
- Das Flugzeug ist symmetrisch bezüglich seiner x - z -Ebene.

Die folgende verkürzte Herleitung der Bewegungsgleichungen ist in [28] detailliert dargestellt und verwendet das flugzeugfeste Koordinatensystem als Bezugskoordinatensystem. Grundlage für die Beschreibung der Flugzeugbewegung ist das Kräfte- und Momentengleichgewicht um den Schwerpunkt. Dabei wird für die translatorische Bewegung der Impulssatz und für die

rotatorische Bewegung der Drehimpulssatz herangezogen. Der Impulssatz lautet:

$$m \cdot \left[\frac{d(\vec{V}_K)}{dt} \right]_f^g = \vec{R}_f^A + \vec{F}_{TW,f} + \vec{G}_f = \left(\sum \vec{F} \right)_f. \quad (4.1)$$

Die in Gleichung 4.1 auftretenden Kräfte unterteilen sich dabei in die Luftkraft \vec{R}_f^A , den Triebwerksschub $\vec{F}_{TW,f}$ und die Gewichtskraft \vec{G}_f . \vec{V}_K bezeichnet den Vektor der Bahngeschwindigkeit, bestehend aus den Komponenten u_K , v_K und w_K . Unter Berücksichtigung der Tatsache, dass \vec{V}_K im flugzeugfesten Koordinatensystem angegeben ist, seine Ableitung aber gegenüber dem erdfesten System erfolgt, ergibt sich für die translatorische Bewegung folgender Ausdruck:

$$\dot{\vec{V}}_{Kf}^f = \frac{1}{m} \cdot \left(\sum \vec{F} \right)_f - \vec{\omega}_f^{gf} \times \vec{V}_{Kf} = \frac{1}{m} \cdot \left(\sum \vec{F} \right)_f - \begin{bmatrix} p_K \\ q_K \\ r_K \end{bmatrix}_f \times \begin{bmatrix} u_K \\ v_K \\ w_K \end{bmatrix}_f. \quad (4.2)$$

Die Größe $\vec{\omega}_f^{gf}$ beschreibt im flugzeugfesten Koordinatensystem den Vektor der Drehgeschwindigkeit des flugzeugfesten gegenüber dem erdfesten System. Für die Beschreibung der rotatorischen Bewegung wird der Drehimpulssatz, auch Drallsatz genannt, verwendet. Dieser beschreibt das Momentengleichgewicht um den Schwerpunkt und kann gemäß Gleichung 4.3 dargestellt werden.

$$\left(\frac{d\vec{L}}{dt} \right)_f^g = \mathbf{T}_f \cdot \left(\frac{d\vec{\omega}_f^{gf}}{dt} \right)_f^g = \sum \vec{M}_f \quad (4.3)$$

Dabei beschreibt \vec{L} den Drehimpuls, $\sum \vec{M}_f$ die Summe aller Momente und \mathbf{T}_f den Trägheitstensor in flugzeugfesten Koordinaten, der für ein symmetrisches Flugzeug wie folgt definiert ist:

$$\mathbf{T}_f = \begin{bmatrix} I_{xx} & 0 & -I_{zx} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix}_f. \quad (4.4)$$

Damit kann Gleichung 4.3 nach den Ableitungen der Drehgeschwindigkeiten umgestellt werden, wobei ähnlich wie zuvor beim Impulssatz die Ableitung bezüglich des erdfesten Koordinatensystems im sogenannten Euler-Term $\vec{\omega}_f^{gf} \times [\mathbf{T}_f \cdot \vec{\omega}_f^{gf}]$ berücksichtigt wird:

$$\dot{\vec{\omega}}_f^{gf} = \mathbf{T}_f^{-1} \cdot \left(\sum \vec{M}_f - \vec{\omega}_f^{gf} \times [\mathbf{T}_f \cdot \vec{\omega}_f^{gf}] \right). \quad (4.5)$$

Damit sind die Zustandsgleichungen für die Geschwindigkeiten und die Drehraten gefunden; für die Bestimmung der Lage des Flugzeuges im Raum werden die Eulerwinkel verwendet. Diese sind der Hängewinkel Φ , der Nickwinkel Θ sowie der Gierwinkel Ψ und können gemäß

Gleichung 4.6 durch Integration und Zuhilfenahme der Transformationsmatrix $\mathbf{M}_{\Phi f}$ aus den Drehraten bestimmt werden.

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \mathbf{M}_{\Phi f} \cdot \vec{\omega}_f^{gf} = \begin{bmatrix} 1 & \sin \Phi \cdot \tan \Theta & \cos \Phi \cdot \tan \Theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi / \cos \Theta & \cos \Phi / \cos \Theta \end{bmatrix}_f \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix}_f \quad (4.6)$$

Die Position eines Flugzeuges relativ zu einem Startpunkt kann schließlich durch Integration der Komponenten des Bahngeschwindigkeitsvektors \vec{V}_{Kf} bestimmt werden. Da die Position häufig im geodätischen Koordinatensystem angegeben wird, ist eine Transformation mit Hilfe der Matrix \mathbf{M}_{gf} erforderlich. Diese Matrix ist gemäß der in [94] dargestellten Regeln zusammengesetzt.

$$\frac{d(\vec{s}_g)}{dt} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_g = \begin{bmatrix} u_K \\ v_K \\ w_K \end{bmatrix}_g = \mathbf{M}_{gf} \cdot \vec{V}_{Kf} = \mathbf{M}_{gf} \cdot \begin{bmatrix} u_K \\ v_K \\ w_K \end{bmatrix}_f \quad (4.7)$$

Damit sind alle nichtlinearen Zustandsdifferentialgleichungen dargestellt, die zur Bestimmung der kinematischen Größen der Flugsimulation benötigt werden und im Block der Flugzeugdynamik implementiert sind. Ebenso stellen diese Gleichungen die Grundlage für jede Reglerauslegung dar, wobei sie bei linearen Regelungsansätzen im Rahmen der dafür nötigen Linearisierung deutlich vereinfacht werden [28].

4.2 Modellannahmen und Umsetzung der Regelungsverfahren

Die kinematischen Größen, die im Block der Flugzeugdynamik berechnet werden, sind je nach Qualität des aerodynamischen Modells und des Schubmodells des zu simulierenden Flugzeuges idealisiert und stehen im realen System so nicht zur Verfügung. Dies liegt unter anderem darin begründet, dass die verwendete Sensorik ein eigenes Übertragungsverhalten aufweist, was meist durch dynamische Systeme erster oder zweiter Ordnung beschrieben werden kann [28]. Zusätzlich werden diese Messungen durch etliche Effekte verfälscht wie zum Beispiel Messrauschen, Sensordrift, Zeitverzögerungen oder Einbaufehler. Aus diesem Grunde werden Sensordynamik und auftretende Fehler in der Simulation berücksichtigt. Dies gilt sowohl für alle Elemente des integrierten Navigationssystems, also Drehraten- und Beschleunigungsmesser sowie den Empfänger des Satellitennavigationssystems als auch für die Sensoren für statischen und Staudruck. Eine genaue Beschreibung der implementierten Sensorfehlermodelle ist in [95] und [96] gegeben. Die Datenfusion und Fehlerkorrektur des simulierten integrierten Navigationssystems basiert auf einem Kalman-Filter, der die Satellitenrohdaten verarbeitet und in [97] detailliert beschrieben ist. Dabei wird in der Simulation die gleiche Algorithmik verwendet, die auch im realen Flugsystem implementiert ist.

Eine weitere Abweichung vom idealen Verhalten tritt dadurch auf, dass Aktuatoren ebenfalls ein eigenes dynamisches Verhalten aufweisen. Dieses muss bei der Reglerauslegung und demzufolge auch in der Simulation berücksichtigt werden. Die Übertragungsfunktionen typischer Servos unbemannter Kleinflugzeuge wurden in [98] ermittelt und sind für die vorliegende Arbeit übernommen worden. Diese sind totzeitbehaftet und ähneln den Ergebnissen aus [52] und [54]. Die gewählte Modellierung des Aktuatorverhaltens ist demnach der verwendeten Klasse von Stellsystemen als angemessen anzusehen.

Da eine der Hauptaufgaben von Regelungssystemen darin besteht, Störeinflüsse auszugleichen, ist es zweckmäßig, auftretende Störungen - soweit bekannt - durch ein Störgrößenmodell in die Reglerauslegung und Simulation mit einzubeziehen [99]. Die maßgeblichen Störungen, die auf ein Flugzeug einwirken, sind Wind und Turbulenz, wobei es auf Grund ihres prinzipiell stochastischen Verhaltens schwierig ist, atmosphärische Störungen exakt zu modellieren [28]. Besonders die angemessene Abbildung atmosphärischer Turbulenz ist wegen ihrer zufällig auftretenden Windböen für die Bewertung von Reglereigenschaften von Bedeutung. Dies gilt in hohem Maße für unbemannte Kleinflugzeuge, da diese unter anderem wegen ihrer geringen Trägheit besonders anfällig für Böen sind. Eine Möglichkeit, für die Reglerbewertung zweckmäßige Turbulenz zu modellieren, besteht darin, Signale zu generieren, die einen ähnlichen Störeinfluss auf das Flugzeug realisieren. Ein Maß hierfür ist das sogenannte Leistungsdichtespektrum, welches die von einem Signal transportierte Leistung in einem betrachteten Frequenzband beschreibt. Durch die Kombination von weißem Rauschen mit geeigneten Filtern ist es möglich, Signale zu erzeugen, die ein ähnliches Leistungsdichtespektrum wie im Flug gemessene Turbulenz aufweisen und somit ähnliche Störungen darstellen. Ein Ansatz hierfür ist das sogenannte *Dryden*-Turbulenzspektrum, das in [28] im Detail beschrieben ist und im Rahmen dieser Arbeit Verwendung findet. Durch ein solches Modell ist es möglich, Störungen zu erzeugen, die sich sehr gut für die Auslegung und Bewertung von Flugreglern eignen.

Die schließliche Implementierung der verschiedenen Regelungsverfahren erfolgt in generischer Form und hat im Sinne der Flugerprobung das übergeordnete Ziel der einfachen Übertragbarkeit der Algorithmen auf die Zielhardware eines Fluggerätes. Generisch bedeutet in diesem Zusammenhang, dass die Reglerarchitekturen für verschiedene Flugzeugtypen und sowohl für bemannte als auch unbemannte Systeme genutzt werden können. Diese vielseitige Anwendbarkeit wird dadurch gewährleistet, dass alle relevanten aerodynamischen Beiwerte und Reglerparameter in separaten Dateien hinterlegt sind und folglich bei einem Wechsel des betrachteten Flugzeuges nur diese geändert zu werden brauchen. Hinsichtlich der Flugerprobung hat sich unter anderem für die verwendeten neuronalen Netze in [100] gezeigt, dass die Umsetzung dafür nötiger C-Coderoutinen mit nicht unerheblichem Zeitaufwand verbunden ist und der zeitnahe Flugversuch somit behindert wird. Daher sind alle Elemente des Regelungssystems so ausgelegt, dass sie mit aktuellen Werkzeugen zur automatischen Codegenerierung kombinierbar sind. Änderungen im Regelungssystem können so in der Simulation validiert und anschließend durch einfache Aktualisierung der Autopilotensoftware im Flug erprobt werden. Dies gilt ebenso für

die neuronalen Netzwerke und die in den Kapiteln 2 und 3 beschriebenen Lernverfahren, was in [101] demonstriert ist. Die in den folgenden Kapiteln vorgestellten Regelungsverfahren sind zur Vorbereitung des Flugversuchs umfangreich validiert und getestet worden. Die modellfreie neuronale Regelung ist dazu unter anderem in [79, 80, 82, 91, 102, 103] und [104] systematisch untersucht und in ihrer Funktionalität sukzessive erweitert worden. Die Regelung mittels dynamischer Inversion ist in [81, 83, 105–107] und [108] in ihrer Funktionalität untersucht und bewertet worden, so dass die Grundlagen für die Flugerprobung als gegeben betrachtet werden können.

5 Modellfreie neuronale Regelung

Beim Entwurf von Regelungssystemen besteht eine grundlegende Frage darin, wieviel Kenntnis dem Entwickler über das Verhalten des realen Systems zur Verfügung steht und inwieweit dieses Wissen in eine dem Ziel angemessene physikalisch-mathematische Beschreibung einfließen soll. Dabei wird das zu regelnde System meist durch geeignete Vereinfachungen von unerwünschten Einflüssen und Rückkopplungen der realen Welt abgeschirmt und in ein in der Komplexität reduziertes mathematisches Modell überführt [99, 109, 110]. Der Ansatz der Systemtheorie erlaubt hierfür eine strukturierte Analyse hinsichtlich Aufbau, Unterteilung in Subsysteme sowie kausaler und temporaler Wirkungsprinzipien [111]. Daraus ableitbare mathematische Prozessmodelle lassen sich in identifizierende und imitierende unterteilen. Identifizierende Modelle nutzen bekannte theoretische Zusammenhänge, meist in Form von Differentialgleichungen, um den Systemzustand möglichst genau abzubilden, weswegen sie auch als Zustandsmodelle bezeichnet werden [9]. Alle Ansätze der klassischen Regelungstechnik beruhen auf solchen Modellen, wobei die genaue Beschreibung des meist nichtlinearen Systemverhaltens eine unter Umständen aufwändige Systemidentifizierung voraussetzt.

Imitierende Modelle hingegen versuchen, die funktionalen Beziehungen zwischen Systemeintritten und -ausgängen bestmöglich abzubilden; sie verzichten auf eine systemtheoretische Analyse und werden aufgrund dieser Eigenschaft auch als Ein-Ausgangsmodelle oder Verhaltensmodelle bezeichnet [9]. Ein imitierendes Modell bietet sich immer dann an, wenn a priori kein ausreichendes Wissen über die physikalischen Zusammenhänge des realen Systems vorliegt, oder es zu aufwändig wäre, dieses über eine Systemidentifizierung zu generieren. Aus dieser Beschreibung sowie den Ausführungen in den Abschnitten 2.4 und 2.5 wird deutlich, dass es sich bei neuronalen Netzen um Verhaltensmodelle handelt, da sie dynamische Systeme nur auf Basis von Ein- und Ausgangsmustern approximieren können. Auch ist es ersichtlich, dass die Synthese eines Regelungssystems nur auf Basis lernfähiger neuronaler Netze generell realistisch ist. Dabei soll der Begriff der Modellfreiheit des im Folgenden vorzustellenden Flugregelungssystems unterstreichen, dass kein aufwändig ermitteltes identifizierendes Modell benötigt wird, wie es in der konventionellen Regelungstechnik erforderlich ist. Damit ist dieser Ansatz der adaptiven Regelung besonders für kleine unbemannte Fluggeräte interessant, da für diese Systeme im Sinne der Entwicklungskosten meist keine umfangreiche Parameteridentifizierung möglich ist. Die für die Basisauslegung des Regelkreises notwendigen Trainingsmuster, bildlich gesprochen also sein Vorwissen, lassen sich vergleichsweise einfach über numerische Simulationen oder aus manuell erfolgten Flugversuchsdaten gewinnen.

5.1 Lernen innerhalb eines Regelkreises

Der Aufbau eines Regelungssystems mit im Betrieb lernfähigen neuronalen Netzwerken bringt auf Grund ihrer Lernfähigkeit einige Vorteile gegenüber der konventionellen linearen Regelung mit sich. Neben dem Aspekt, dass kein hochgenaues aerodynamisches Modell, etwa aus einer Systemidentifizierung, zur Reglerauslegung nötig ist, sind dies folgende:

1. Man erhält ein arbeitspunktunabhängiges, nichtlineares Regelungssystem; es ist nicht notwendig, Reglerparameter für verschiedene Arbeitspunkte zu ermitteln und aufeinander abgestimmt zu optimieren. Im Nominalfall passt sich der neuronale Regler selbstständig an die verschiedenen Flugbereiche und Windsituationen an.
2. Das Fluggerät erhält über den adaptiven Regler eine bessere Toleranz in Extremsituationen. Dies bezieht sich zum einen auf Flugzustände mit starkem Wind und Böen, was besonders für kleinere unbemannte System relevant ist; zum anderen gilt dies bis zu einem bestimmten Grad für Systemschäden und wegen der parallelen Datenverarbeitung im KNN nicht zuletzt für Sensorfehler.

Damit diese potentiellen Vorteile eines lernfähigen Regelungssystems auch realisiert werden können und somit während der Einsatzphase stabil hinzugelernt werden kann, muss eine solide Bewertung der Regeleigenschaften erfolgen. Dies ist von großer Bedeutung, da durch diese Bewertung das Fehlersignal für das KNN generiert und somit die Gewichtsveränderung maßgeblich bestimmt wird. Der allgemeine Aufbau eines solchen neuronalen Regelkreises für ein vollautomatisches unbemanntes Fluggerät ist in Abbildung 5.1 gegeben.

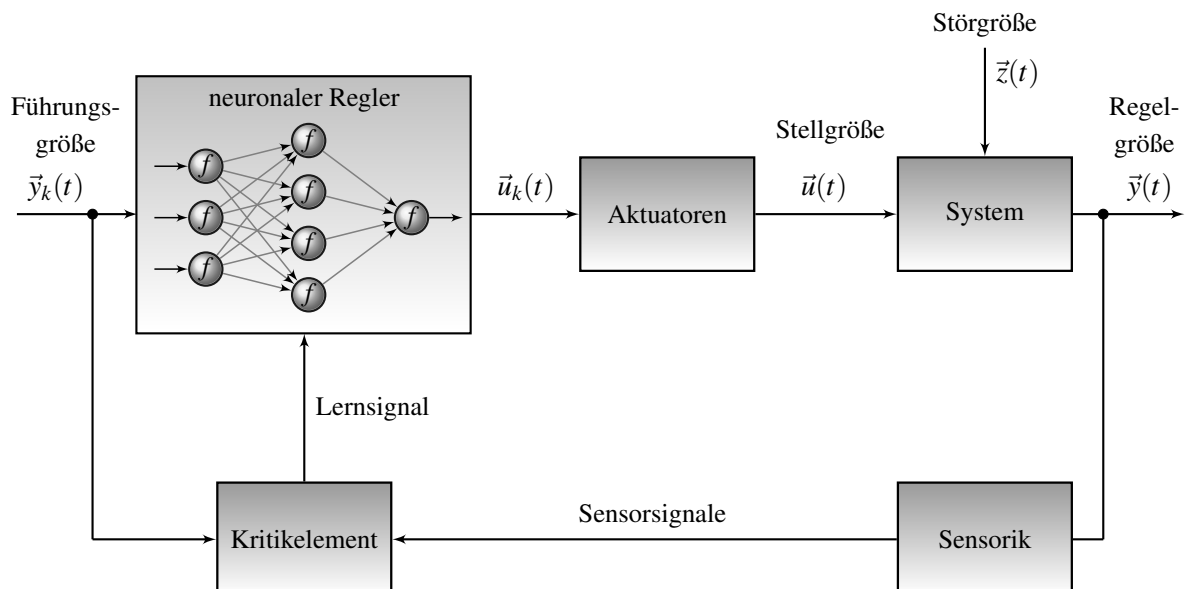


Abbildung 5.1: Allgemeine Darstellung eines lernfähigen Flugregelkreises mit Kritikelement.

Betrachtet man die Komponenten des Regelkreises aus Abbildung 5.1, lassen sich deutliche Parallelen zum Konzept des rationalen Agenten aus der künstlichen Intelligenz und der Robotik

ziehen. Demnach ist ein Agent zunächst einmal ein System, das über geeignete Sensorik seine Umwelt und ihre Einflüsse (in Form von Störungen) wahrnimmt und diese über Aktuatoren selbsttätig beeinflusst. Dabei können die eigenen Aktionen wahrgenommen werden, allerdings nicht unbedingt ihre Wirkung [59]. Übertragen auf ein unbemanntes Fluggerät kann man feststellen, dass es über ein integriertes Navigationssystem und eventuelle Zusatzsensorik eine gewisse Wahrnehmung der Situation erhält und basierend auf diesen Informationen durch seine Ruderkommandos und den Schub in der Umwelt agiert. Rational ist ein Agent dann, wenn er für eine bestimmte Wahrnehmungssituation eine Aktion wählt, die eine Leistungsmaximierung zur Folge hat und er darüber hinaus in der Lage ist, aus seiner Wahrnehmung zu lernen [59]. Auch diese Annahmen gelten für den Flugregelkreis aus Abbildung 5.1, sofern sichergestellt ist, dass die Lernsignale des Kritikelementes der Situation angemessen sind.

Es wird deutlich, dass eine objektive Leistungsbewertung erforderlich ist, damit die Aktionen des Agenten anhand eines Gütekriteriums analysiert und adäquate Lernsignale daraus abgeleitet werden können. Dieser Vorgang erfolgt im Kritikelement, welches den aktuellen Flugzustand gegenüber den Zielen des Flugplanes bewertet und daraus die nötigen Lernsignale für den neuronalen Regler generiert. Der systematische Entwurf des Kritikelementes hat in einem lernfähigen Flugregler also hohe Relevanz, da es ein sicherheitskritisches System darstellt und bei einer Fehlfunktion letztlich auch zum Verlust des Luftfahrzeuges führen kann. Aus diesen Ausführungen ist außerdem ersichtlich, dass vollautomatische lernfähige unbemannte Fluggeräte in vielerlei Hinsicht den gleichen Anforderungen wie in der Robotik Genüge tun müssen, um sich in Richtung echter Systemautonomie zu entwickeln.

5.2 Aufbau des modellfreien neuronalen Flugreglers

Die Auslegung eines modellfreien neuronalen Regelkreises soll im Folgenden am Beispiel der Seitenbewegung eines unbemannten Fluggerätes erläutert werden. Der vorgestellte Ansatz wird auch als indirekte adaptive neuronale Regelung bezeichnet (engl. *indirect adaptive neuro-control*) und ist in Varianten für verschiedene Regelungsanwendungen umgesetzt worden, unter anderem für die Seitenbewegung eines Lenkflugkörpers [63], für Elektromotoren unter Berücksichtigung nichtlinearer Reibungseffekte [17] oder ganz allgemein für nichtlineare Systeme höherer Ordnung [15, 64]. Da es sich um einen adaptiven Regelungsansatz handelt, der nicht auf die Luftfahrtanwendung beschränkt ist, soll die Architektur zunächst in allgemeiner Form gemäß Abbildung 5.2 dargestellt werden.

Neben dem Referenzmodell, das aus der Führungsgröße $\vec{y}_k(t)$ eine Solltrajektorie $\vec{y}_r(t)$ generiert, besteht der Regelkreis aus zwei neuronalen Netzen: dem neuronalen Regler und dem Beobachter. Das in Abschnitt 5.1 eingeführte Kritikelement besteht aus der Kombination aus Referenzmodell und neuronalem Beobachter, der gemäß Abschnitt 2.5 das Übertragungsverhalten des Systems von Stellgröße $\vec{u}(t)$ zur Regelgröße $\vec{y}(t+1)$ abbildet und dessen genaue Funktionsweise im Folgenden erläutert werden soll. Dazu sei angemerkt, dass sich der

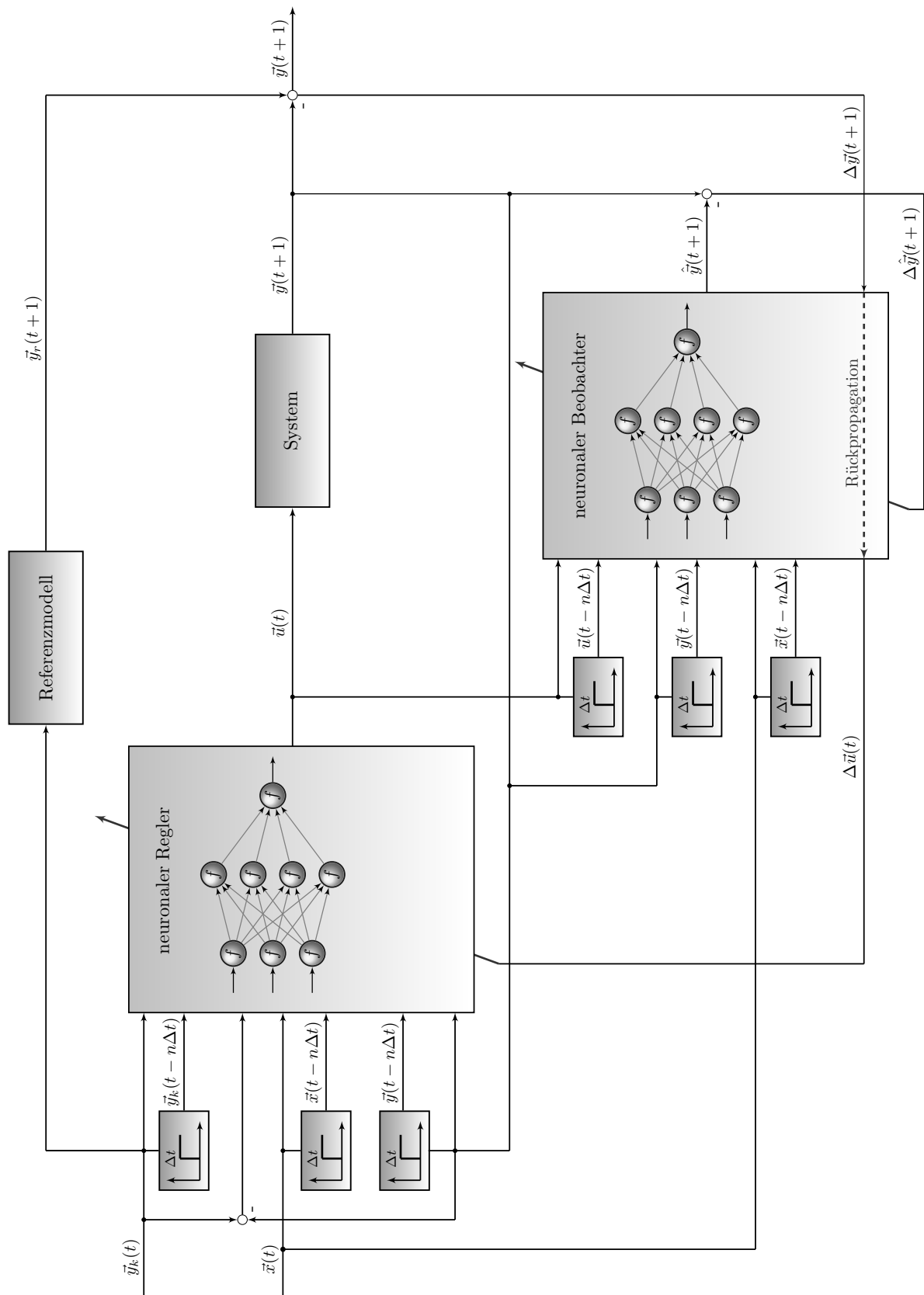


Abbildung 5.2: Darstellung des modellfreien neuronalen Flugregelkreises.

Zeitindex $(t + 1)$ an der für diese Regelung etablierten Notation orientiert [15–17] und lediglich den bereits erläuterten notwendigen zeitlichen Versatz der Schätzung des neuronalen Beobachters um einen Zeitschritt in die Zukunft beschreibt. Im Regelkreis kann der Schätzvorgang des Beobachters auch einen Zeitschritt in die Vergangenheit verlegt werden, um die aktuelle Ausgabe $\vec{y}(t) \approx \hat{\vec{y}}(t)$ zu bestimmen, was in [112] im Detail beschrieben ist.

Das Reglernetzwerk wird trainiert, um den nichtlinearen funktionalen Zusammenhang zwischen Führungsgröße $\vec{y}_k(t)$, Regelgröße $\vec{y}(t)$ und der zur Einhaltung der Solltrajektorie nötigen Stellgröße $\vec{u}(t)$ abzubilden.

$$f_{RN} : \{\vec{y}_k(t), \vec{y}(t)\} \rightarrow \vec{u}(t) \quad (5.1)$$

Dadurch wird, wie bereits erwähnt, ein nichtlineares Regelgesetz implementiert, das selbstständig auf Veränderungen des Arbeitspunktes reagiert. Daneben ist ersichtlich, dass der neuronale Regler (ebenso wie der Beobachter) gemäß Abschnitt 2.5 auch Zustandsgrößen und von ihnen abgeleitete Signale, zusammengefasst im Vektor \vec{x} , als Eingaben erhält. Bezogen auf die Generierung eines stabilen Lernsignals, das es dem neuronalen Regler erlaubt hinzuzulernen, ergibt sich allerdings die Problematik, dass meist nur eine auftretende Regelabweichung $\Delta\vec{y}(t)$ messtechnisch erfassbar ist, während sich der damit verbundene Fehler in der Stellgröße $\Delta\vec{u}(t)$ nicht ohne weiteres ermitteln lässt. Um $\Delta\vec{u}(t)$ und damit eine Fehlergröße zum Training des Reglernetzwerkes gemäß Gleichung 2.10 zu ermitteln, wird das Beobachternetzwerk genutzt, das folgende nichtlineare Funktion abbildet (siehe Abschnitt 2.5):

$$f_{BN} : \{\vec{u}(t), \vec{y}(t)\} \rightarrow \hat{\vec{y}}(t + 1) . \quad (5.2)$$

Dadurch, dass die Ausgabe des Beobachters $\hat{\vec{y}}(t + 1)$ mit der tatsächlichen Ausgabe des zu regelnden Systems $\vec{y}(t + 1)$ verglichen werden kann, erhält man das Lernsignal $\Delta\hat{\vec{y}}(t + 1)$ des Beobachternetzwerkes. Das stetige Lernen des neuronalen Beobachters ermöglicht folglich eine genaue Modellierung der Dynamik der Regelstrecke, wodurch letztlich auch der nichtlineare Zusammenhang zwischen Stellgröße $\vec{u}(t)$ und Ausgabegröße $\vec{y}(t + 1) \approx \hat{\vec{y}}(t + 1)$ abgebildet wird. Dieses in Echtzeit trainierte neuronale Systemmodell kann nun genutzt werden, um die Regelabweichung $\Delta\vec{y}(t + 1)$ nach den Rechenvorschriften aus Abschnitt 2.3.1 rückwärts durch das Beobachternetzwerk zu propagieren. Dazu wird die Regelabweichung in Gleichung 2.16 eingesetzt und so der rückpropagierte Fehler $\delta_{BN}^{(L)}$ der Ausgabeschicht des Beobachternetzwerkes bestimmt.

$$\delta_{BN}^{(L)} = -\Delta\vec{y}(t + 1) \quad (5.3)$$

Setzt man $\delta_{BN}^{(L)}$ nun in Gleichung 2.19 ein, können die Vektoren der rückpropagierten Fehler $\vec{\delta}^{(l)}$ für jede Schicht des Beobachternetzes bestimmt werden. Dadurch, dass in der Eingabeschicht aller verwendeter KNN die Identitätsfunktion als Transferfunktion verwendet wird und die Stellgröße $\vec{u}(t)$ ein Element des Eingabevektors $\vec{x}(t)$ des Beobachters ist, lässt sich folgender funktionaler Zusammenhang finden:

$$\Delta\vec{u}(t) = f(\vec{\delta}_{BN}^{(1)}) . \quad (5.4)$$

Man erkennt, dass durch Rückpropagation des Regelfehlers $\Delta\vec{y}(t+1)$ durch den neuronalen Beobachter der für die Regelabweichung verantwortliche Fehler in der Stellgröße $\Delta\vec{u}(t)$ als eine Funktion des rückpropagierten Fehlers $\vec{\delta}_{BN}^{(1)}$ bestimmbar ist. Durch das Lernsignal $\Delta\vec{u}(t)$ kann die Ausgabe des neuronalen Reglers $\vec{u}(t)$ derart beeinflusst werden, dass eine Minimierung der Regelabweichung $\Delta\vec{y}(t)$ erreicht wird. Die genaue Bestimmung von $\Delta\vec{u}(t)$ wird in Abschnitt 5.2.3 beschrieben. Es lässt sich ergänzend feststellen, dass das Beobachternetzwerk durch diese Form der Rückpropagation zur Abbildung der invertierten Jacobi-Matrix des zu regelnden Systems \mathbf{J}_S^{-1} genutzt wird [17, 63, 112]. Die Jacobi-Matrix \mathbf{J}_S beschreibt allgemein die Änderung der m Ausgangsgrößen eines Systems bei sich ändernden n Eingängen. Damit stellt sie eine Linearisierung des funktionalen Zusammenhanges zwischen Ein- und Ausgabegrößen um einen aktuellen Arbeitspunkt dar und kann in folgender Form angegeben werden:

$$\mathbf{J}_S = \frac{\Delta\vec{y}}{\Delta\vec{u}} = \begin{bmatrix} \frac{\partial y_1}{\partial u_1} & \frac{\partial y_1}{\partial u_2} & \dots & \frac{\partial y_1}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial u_1} & \frac{\partial y_m}{\partial u_2} & \dots & \frac{\partial y_m}{\partial u_n} \end{bmatrix}. \quad (5.5)$$

Aus Gleichung 5.6 wird deutlich, dass der neuronale Beobachter durch die Rückpropagation des Regelfehlers $\Delta\vec{y}(t)$ die Funktion der inversen Jacobi-Matrix erfüllt, ohne dass die mathematischen Zusammenhänge der partiellen Ableitungen $\partial y_m / \partial u_n$ bekannt zu sein brauchen und ohne dass das Problem der Invertierbarkeit der Matrix behandelt zu werden braucht.

$$\Delta\vec{u} = \Delta\vec{y} \cdot \mathbf{J}_S^{-1}. \quad (5.6)$$

Darüber hinaus ist festzustellen, dass durch das stetige Lernen des neuronalen Beobachters auch die Abbildung der inversen Jacobi-Matrix für jeden neuen Arbeitspunkt ermöglicht wird. Eine genaue Herleitung der beschriebenen Abbildungseigenschaften und eine Betrachtung der Stabilität sind in [112] dargestellt.

Die Nutzung des neuronalen Beobachters innerhalb des Kritikelementes zeigt, dass die Abbildung der Dynamik des zu regelnden Systems stabil erlernt werden muss. Gleichzeitig ist es, gerade bei abrupten Verschiebungen des Arbeitspunktes, zum Beispiel durch starken Wind oder Beschädigungen am System, wichtig, dass diese Änderungen auch mit einer hohen Konvergenzgeschwindigkeit erlernt werden können, da ansonsten die Qualität des Lernsignals stark reduziert ist. Somit sind die in Kapitel 3 vorgestellten robusten und schnellen Lernverfahren für die Bestimmung eines verlässlichen Lernsignals $\Delta\vec{u}(t)$ von großer Bedeutung.

5.2.1 Wahl der Regelgröße zur Ermittlung des Lernsignals

Aus Abbildung 5.2 und den vorigen Ausführungen wird deutlich, dass zur Regelung der Seitenbewegung eines Flugzeuges eine Regelgröße verwendet werden muss, die die Ermittlung eines stabilen Lernsignals ermöglicht. Um also die laterale Bahnfolge angemessen regeln zu können, muss die Sollbahn so definiert werden, dass eine stetige Messung der seitlichen Abweichung

möglich ist. Besonders für unbemannte Flugsysteme bietet sich dafür die Bahnführung auf kubischen Spline-Kurven an, wie es in [113] und [114] vorgeschlagen wird. Eine solche Flugbahn besteht aus einer Vielzahl einzelner Kurvenstücke, die ihrerseits als Polynome dritten Grades über vier Punkte P_0 - P_3 definiert sind; ein Beispiel einer Spline-Kurve ist in Abbildung 5.3 gegeben.

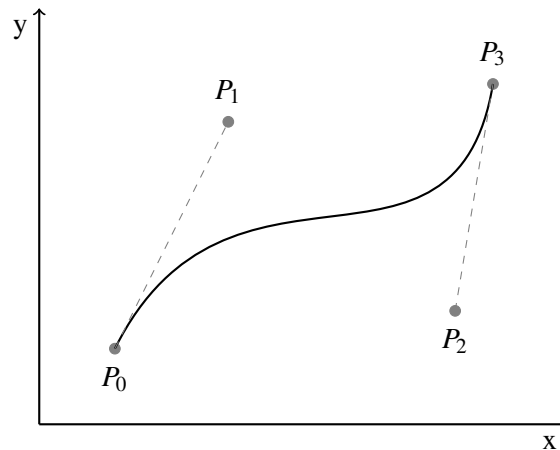


Abbildung 5.3: Darstellung einer kubischen Spline-Kurve.

Betrachtet man die Punkte P_0 - P_3 in ihren x - y -Koordinaten im geodätischen System, lassen sich zur Berechnung der aktuellen Position auf einem Kurvenabschnitt folgende zwei Gleichungen finden:

$$x(t) = a_3 \cdot t^3 + a_2 \cdot t^2 + a_1 \cdot t + x_0, \quad (5.7)$$

$$y(t) = b_3 \cdot t^3 + b_2 \cdot t^2 + b_1 \cdot t + y_0. \quad (5.8)$$

Hierbei ist t ein Laufparameter mit $0 \leq t \leq 1$, der die bereits durchschrittene Kurvenlänge markiert. Die Koeffizienten $a_1 = 3(x_1 - x_0)$, $a_2 = 3(x_0 - 2x_1 + x_2)$ sowie $a_3 = -x_0 + 3x_1 - 3x_2 + x_3$ werden mit Hilfe der x -Koordinaten bestimmt, die Koeffizienten b_1 bis b_3 nach gleicher Rechenvorschrift unter Verwendung der Koordinaten y_0 bis y_3 . Weiterhin ist aus Abbildung 5.3 zu erkennen, dass durch die relative Lage der Stützpunkte P_1 und P_2 die Steigung der Tangenten an Start- und Endpunkt definiert ist. Durch gleiche Wahl der Steigung an End- und Anfangspunkt aufeinander folgender Splines kann folglich ein glatter Kurvenübergang gewährleistet werden. Darüber hinaus sind die Kurvenstücke zweimal stetig differenzierbar, wodurch überall auf dem Spline die Berechnung der Krümmung möglich ist [62]. Damit können aus der bekannten Geometrie der Sollbahn und flugmechanischen Zusammenhängen zusätzliche Vorsteuergrößen abgeleitet werden, die als weitere Informationen zur Regelung nutzbar sind [113, 115].

Durch diese mathematische Beschreibung der Sollbahn kann nun mit Hilfe eines integrierten Navigationssystems, wie es in [97] beschrieben ist, in jedem Regelungsschritt die aktuelle Position und damit die seitliche Bahnabweichung d bestimmt werden. Diese wird im Englischen

auch als *cross track error* bezeichnet und ist rechts von der Bahn positiv definiert. Mit der Bahnabweichung d ist eine globale Qualitätsgröße gegeben, die es vom neuronalen Regler zu minimieren gilt. Diese kann prinzipiell auch direkt in Gleichung 5.3 eingesetzt und zur Rückpropagation durch den Beobachter verwendet werden, was hinsichtlich der Ergebnisse in [103] dargestellt ist. Um bei der Einhaltung der Solltrajektorie allerdings auch eine gewünschte Dynamik vorzugeben, ist gemäß Abbildung 5.2 sowie [63] und [17] die Verwendung eines Referenzmodells sinnvoll. Da die seitliche Bahnablage gegen Null gehen soll, ist diese Führungsgröße nicht als Eingang eines dynamischen Referenzmodells geeignet, so dass eine von d abgeleitete Größe gewählt wird. Dazu wird in Anlehnung an [28] die Relativposition in Bezug zur Sollbahn nach der Zeit abgeleitet; durch erneute Ableitung kann so eine kommandierte Querschleunigung $a_{y,k}$ als Führungsgröße generiert werden, die durch das Referenzmodell in eine Solltrajektorie $a_{y,r}$ überführt wird [82]. Durch Vergleich der gemessenen Beschleunigung a_y mit dem Sollwert $a_{y,r}$ kann der Regelfehler Δa_y bestimmt und zur Rückpropagation durch den Beobachter genutzt werden. Damit kann auf Basis der gemessenen Bahnabweichung d ein stabiles Lernsignal Δu ermittelt werden, mit dem der neuronale Regler diese zu minimieren vermag.

5.2.2 Auslegung des neuronalen Reglers

Nachdem das Grundprinzip des lernfähigen Regelkreises, die Idee des Kritikelementes und die Wahl des Lernsignals erläutert worden sind, soll der konkrete Aufbau des neuronalen Reglers beschrieben werden. Da in Kapitel 2 dargelegt wurde, dass vorwärtsgerichtete Netzwerke mit einer, maximal zwei verdeckten Schichten verwendet werden, verbleibt nun, die Wahl der Eingangssignale, respektive der Ausgangsgröße des Reglers zu treffen. Zum Ausgangssignal des Reglers kann festgestellt werden, dass die generierte Stellgröße das Flugzeug so genau wie möglich auf einer gemäß Abschnitt 5.2.1 definierten Flugbahn halten soll. Nach [28] ist ein Querruderausschlag ξ der effizienteste Weg, die Seitenbewegung zu steuern, da so eine Querneigung aufgebaut und in Folge der Auftriebsvektor geschwenkt wird, was das Einleiten einer Kurve zur Folge hat. Durch einen Ausschlag des Querruders $\delta\xi$ wird die Rollbewegung gesteuert und führt zu der Wirkungskette $\delta p, \delta\Phi, \delta\chi$ sowie schließlich zu δd , was genau der messbaren Qualitätsgröße der Bahnablage aus Abschnitt 5.2.1 entspricht und eine direkte Steuerung der Flugbahn erlaubt. Darüber hinaus werden gerade kleine unbemannte Fluggeräte wegen des geringeren Fertigungsaufwandes häufig ohne Seitenruder konstruiert, so dass die Ausgangsgröße des neuronalen Reglers also gewählt wird zu:

$$y_{RN} = \xi . \quad (5.9)$$

Bezüglich der Eingangssignale des neuronalen Reglers sind in den Abschnitten 2.5 und 5.2 bereits allgemeine Aussagen getroffen worden, die kombiniert mit der Wahl der Führungsgröße $a_{y,k}$ zu der in Abbildung 5.4 dargestellten zeitdiskreten Verschaltung des Reglernetzwerkes führen. Dabei wird die messbare Regelabweichung $\Delta a_{y,k}$ zwischen Führungsgröße $a_{y,k}$ und Regelgröße a_y als zusätzliches Eingangssignal genutzt.

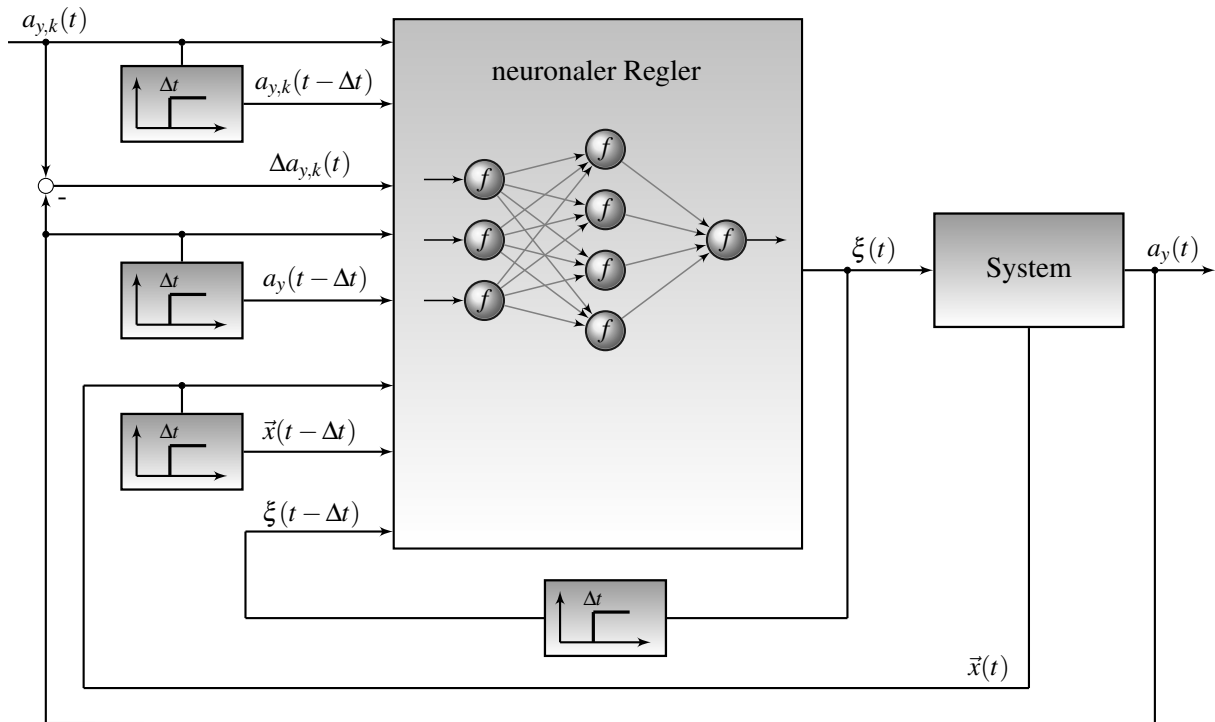


Abbildung 5.4: Verschaltung des neuronalen Reglers.

Nach [63] sollte ein neuronaler Regler sinnvollerweise mindestens die gleichen Eingabegrößen wie ein konventionelles, für die gleiche Aufgabe ausgelegtes Regelungssystem aufweisen. In Abschnitt 2.5 ist außerdem die Nutzung verfügbarer Zustandsgrößen \vec{x} als zusätzliche Informationsquelle empfohlen worden, so dass sich unter Einbeziehung dieser Randbedingungen folgender Eingangsvektor des Reglernetzwerkes ergibt:

$$\vec{x}_{RN} = [a_{y,k}(t), a_{y,k}(t - \Delta t), \Delta a_{y,k}(t), a_y(t), a_y(t - \Delta t), \vec{x}(t), \vec{x}(t - \Delta t), \xi(t - \Delta t)]^T. \quad (5.10)$$

Um die Darstellung nicht unnötig zu komplizieren, enthält der Vektor \vec{x} aus Gleichung 5.10 neben den für die Regelungsaufgabe relevanten Zuständen auch von ihnen abgeleitete Größen. Dazu gehören unter anderem der vorgesteuerte Hängewinkel Φ_k und der vorgesteuerte Bahnazimut χ_k , deren Berechnung von der aktuellen Relativposition zur Zielbahn und ihrer bekannten Geometrie abhängt, was in [113] näher erläutert ist. Diese Größen werden deswegen als vorgesteuert bezeichnet, weil in ihre Bestimmung die Kenntnis der zukünftig zu durchfliegenden Spline-Geometrie eingeht, was sie zu einer sinnvollen zusätzlichen Trainingsinformation macht. Alle Elemente des Vektors \vec{x} , die potentiell weitere Informationen für den Lernprozess beinhalten, sind in Tabelle 5.1 zusammengestellt. Setzt man alle aufgelisteten Größen in Gleichung 5.10 ein, wird deutlich, dass der Eingangsvektor des Reglers \vec{x}_{RN} mit 22 Elementen eine beachtliche Größe erreicht; gleichwohl stellen diese Signale mögliche Informationsquellen für den Lernprozess dar, die es zunächst zu registrieren gilt. Eine Detailauswahl der Eingangssignale kann schließlich auf Grundlage des in Abschnitt 2.7 beschriebenen Entwurfsverfahrens erfolgen.

| Bezeichnung | Symbol | Einheit |
|----------------------------|----------|---------|
| Rollrate | p | rad/s |
| Gierrate | r | rad/s |
| Hängewinkel | Φ | ° |
| vorgesteuerter Hängewinkel | Φ_k | ° |
| Bahnazimut | χ | ° |
| vorgesteuerter Bahnazimut | χ_k | ° |
| Bahnablage | d | m |

Tabelle 5.1: Zusammensetzung des Vektors \vec{x} .

5.2.3 Auslegung des neuronalen Beobachters

Das Beobachternetzwerk ist neben dem Referenzmodell Bestandteil des zuvor eingeführten und beschriebenen Krtikelementes. Wie bereits in Abschnitt 5.2 erläutert, erlernt es das dynamische Verhalten des zu regelnden Systems, um aus der Regelabweichung durch Rückpropagation ein Lernsignal für den neuronalen Regler zu bestimmen. Um dies zu ermöglichen, muss der Beobachter die Dynamik der Regelgröße approximieren, so dass seine Ausgangsgröße gemäß Gleichung 5.11 gewählt wird.

$$y_{BN} = \hat{a}_y \quad (5.11)$$

In Abschnitt 2.5 ist dargelegt, dass ein vorwärtsgerichtetes KNN die Ausgabe eines dynamischen Systems beliebiger Ordnung zu schätzen vermag, solange es aktuelle und zeitverzögerte Ein- und Ausgangsgrößen des abzubildenden Systems als Eingabesignale erhält. Darüber hinaus ist in [112] dargestellt, dass das Beobachternetzwerk auch einen Zeitschritt Δt in die Vergangenheit versetzt werden kann, um die Ausgabe des aktuellen Zeitschrittes $\vec{y}(t)$ zu schätzen, in diesem konkreten Falle also $\hat{a}_y(t)$. Kombiniert man diese Annahmen, kann folgender Eingangsvektor des neuronalen Beobachters \vec{x}_{BN} zusammengestellt werden:

$$\vec{x}_{BN} = [a_y(t - \Delta t), a_y(t - n\Delta t), \vec{x}(t - \Delta t), \vec{x}(t - n\Delta t), \xi(t - \Delta t), \xi(t - n\Delta t)]^T. \quad (5.12)$$

Dabei ist der Vektor \vec{x} aus Gleichung 5.12 wie in Tabelle 5.1 beschrieben zusammengesetzt. Der maximal zu wählende Wert für n richtet sich nach der Ordnung des abzubildenden Systems. Im Sinne einer steigenden Zahl von Elementen im Vektor \vec{x}_{BN} sollte ein Kompromiss zwischen Abbildungsgenauigkeit und Netzwerkgröße gefunden werden. Eine genauere Analyse, auch hinsichtlich der Nutzung der im Vektor \vec{x} zusammengefassten Größen, kann gemäß des Entwurfsprozesses aus Abschnitt 2.7 erfolgen, wobei zunächst für n ein Maximalwert von 3 angenommen wird. Analog zum Reglernetzwerk ist in Abbildung 5.5 die Verschaltung des Beobachternetzes innerhalb des Regelkreises dargestellt. Anhand der in dieser Detaildarstellung zusätzlich vorhandenen Zeitverzögerungen wird deutlich, dass der Beobachter ausschließlich Werte aus der Vergangenheit nutzt, um den aktuellen Systemausgang zu schätzen.

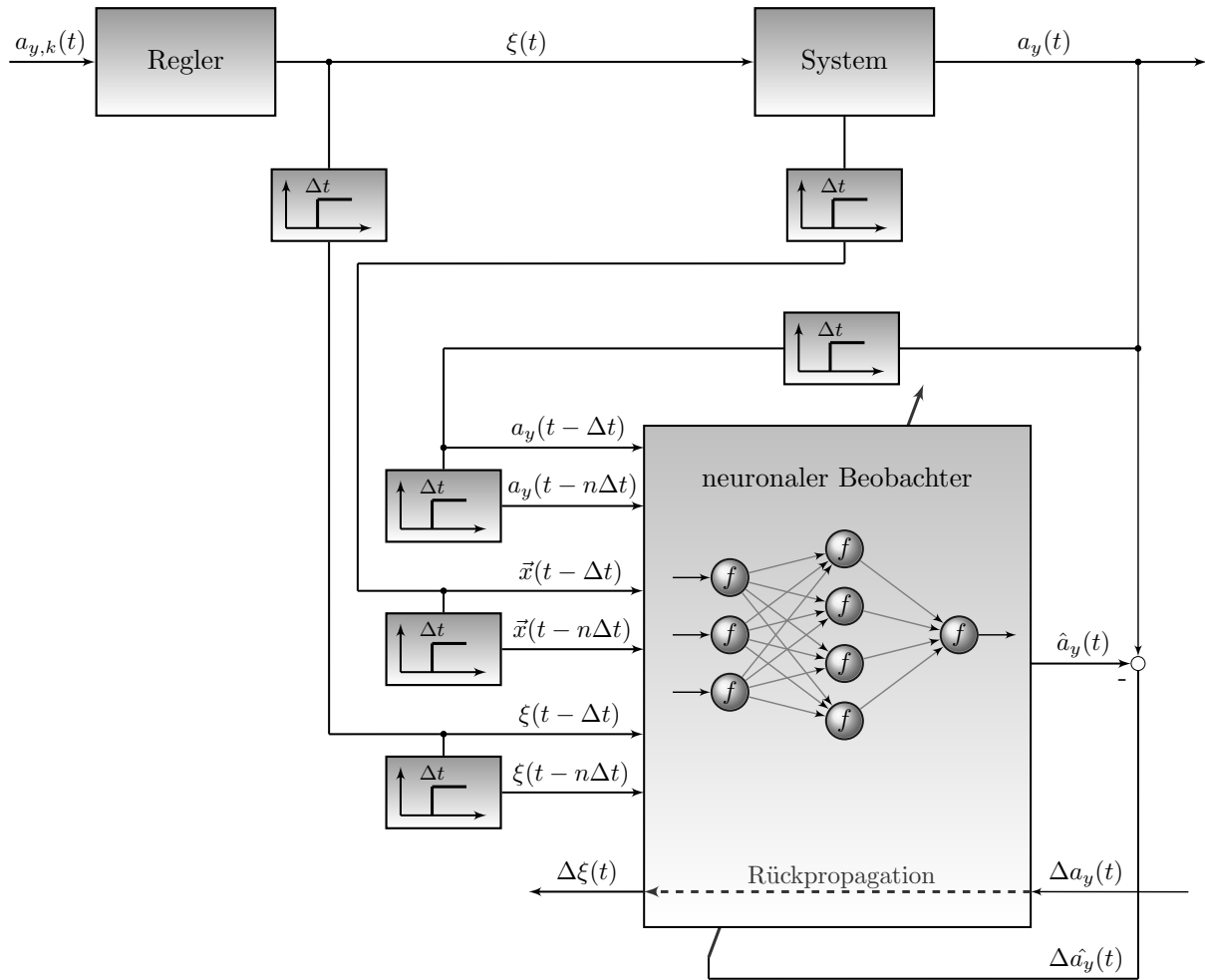


Abbildung 5.5: Verschaltung des neuronalen Beobachters.

In Abschnitt 5.2 ist die Grundidee der Rückpropagation des Regelfehlers durch den Beobachter bereits umrissen worden. Für die Querbeschleunigung als Führungsgröße ergibt sich folgender Zusammenhang für die Regelabweichung von der durch das Referenzmodell vorgegebenen Wunschkurve $a_{y,r}$:

$$\Delta a_y = a_{y,r} - a_y. \quad (5.13)$$

Diese Regelabweichung kann nun in Gleichung 5.3 eingesetzt werden, so dass man den rückpropagierten Fehler der Ausgangsschicht L des Beobachternetzwerkes $\delta_{BN}^{(L)}$ in Abhängigkeit von Δa_y erhält.

$$\delta_{BN}^{(L)} = -\Delta a_y \quad (5.14)$$

Dieser Zwischenschritt ist nun Grundlage für die weitere Rückpropagation nach Gleichung 2.19. Geht man von einem dreischichtigen Netzwerk mit Identitätsfunktion in der Eingangsschicht aus, lassen sich folgende Zusammenhänge finden:

$$\vec{\delta}_{BN}^{(2)} = -\text{diag}(\vec{f}'^{(2)}) \cdot \mathbf{w}^{(2)} \cdot \Delta a_y, \quad (5.15)$$

$$\vec{\delta}_{BN}^{(1)} = - \mathbf{w}^{(1)} \cdot \text{diag} \left(\vec{f}'^{(2)} \right) \cdot \mathbf{w}^{(2)} \cdot \Delta a_y . \quad (5.16)$$

Aus den Gleichungen 5.15 und 5.16 wird deutlich, dass alle Größen zur Berechnung von $\vec{\delta}_{BN}^{(1)}$ aus der Vorwärtspropagation zur Bestimmung von \hat{a}_y bekannt sind. Die allgemeine Definition des rückpropagierten Fehlers δ_j eines Neurons j nach Gleichung 2.14 besagt nun, dass er die Änderung der Fehlerfunktion E bei sich ändernder Aktivität n_j beschreibt. Wegen der Identitätsfunktion in der ersten Schicht ist die Aktivität aber gleich der Eingabe: $n_j = \underline{x}_j$. Zur Verdeutlichung lässt sich der rückpropagierte Fehler $\vec{\delta}^{(1)}$, der also die Änderung der Fehlerfunktion E bei sich änderndem Eingabevektor \underline{x} beschreibt, damit in Komponentenschreibweise darstellen:

$$\vec{\delta}^{(1)} = \left[\frac{\partial E}{\partial \underline{x}_1}, \frac{\partial E}{\partial \underline{x}_2}, \dots, \frac{\partial E}{\partial \underline{x}_q} \right] . \quad (5.17)$$

Geht man nun davon aus, dass das Querrudersignal ξ das n -te Element des Eingabevektors \underline{x} ist, kann man für den rückpropagierten Fehler dieses Eingabeneurons festhalten, dass:

$$\vec{\delta}_{n,BN}^{(1)} = \frac{\partial E}{\partial \xi} . \quad (5.18)$$

Da die Fehlerfunktion E im Rahmen der Rückpropagation durch das Beobachternetzwerk aber aus Δa_y gewonnen wurde, besagt das Signal $\vec{\delta}_{n,BN}^{(1)}$, dass sich ξ um $\delta \xi \triangleq \Delta \xi$ ändern muss, damit sich a_y um Δa_y ändert. Man erkennt, dass in Anlehnung an Gleichung 5.4 der Stellgrößenfehler $\Delta \xi$ in der Tat eine Funktion des rückpropagierten Fehlers $\vec{\delta}_{BN}^{(1)}$ der ersten Schicht ist und somit das Lernsignal für den neuronalen Regler bestimmt werden kann.

5.2.4 Beschreibung des Referenzmodells

Referenzmodelle werden generell genutzt, um einem geschlossenen Regelkreis eine bestimmte Wunschkurve aufzuprägen und kommen dabei nicht nur im Rahmen der neuronalen Regelung zur Anwendung [16, 52, 63, 99]. Ein solches Modell kann als stark abstrahiertes Modell des zu regelnden Prozesses aufgefasst werden. Es wird, wie in Abbildung 5.2 dargestellt, verwendet, um aus der Führungsgröße $\vec{y}_k(t)$ die Solltrajektorie $\vec{y}_r(t)$ zu bilden, die mit der tatsächlichen Regelgröße $\vec{y}(t)$ verglichen werden kann. Es erfüllt als Bestandteil des Kritelementes damit die Aufgabe, den Lernerfolg des neuronalen Reglers anhand der Ausgangsgröße zu bewerten. Man kann das Referenzmodell auch als Kommandofilter bezeichnen, da die Führungsgröße in ihren Signaleigenschaften in einen Bereich gebracht wird, dem das reale System zu folgen vermag. Wegen der einfachen Implementierung werden in der Regel lineare Referenzmodelle genutzt; in Anlehnung an [17] und [63] wird hier ein lineares System zweiter Ordnung verwendet, wie in Abbildung 5.6 ersichtlich. Die Wahl eines schwingungsfähigen Systems ist sinnvoll, da die seitliche Bahnfolge geregelt werden soll - die Forderung nach Stabilität des Referenzmodells ist offensichtlich.

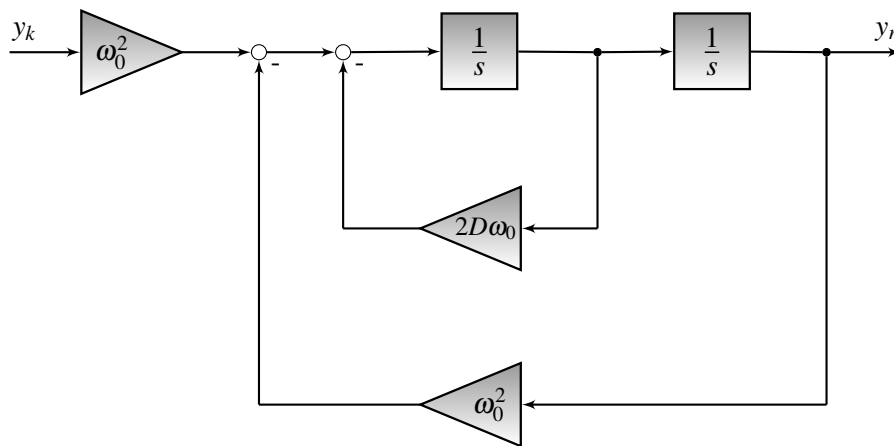


Abbildung 5.6: Lineares Referenzmodell zweiter Ordnung.

Die Wahl der Parameter D und ω_0 richtet sich nach der Dynamik des zu regelnden Systems, da das Referenzmodell keine schnellere Dynamik als die Regelstrecke aufweisen darf, um keinen destabilisierenden Effekt zu haben. Die Übertragungsfunktion des Referenzmodells kann angegeben werden mit:

$$F(s) = \frac{1}{1 + \frac{2D}{\omega_0}s + \frac{1}{\omega_0^2}s^2} . \quad (5.19)$$

5.3 Erweiterung des Regelkreises

Um das Verhalten des geschlossenen Regelkreises qualitativ zu steigern, soll in Anlehnung an [17] ein erweitertes Referenzmodell eingeführt werden, mit dessen Hilfe sich ein verbessertes Lernsignal für das Training des neuronalen Reglers bestimmen lässt. Aus der Verschaltung im Regelkreis gemäß Abbildung 5.7 ist erkennbar, dass das erweiterte Referenzmodell den Regelfehler $\Delta \vec{y}(t+1)$ und das Referenzsignal $\vec{y}_r(t+1)$ als Eingangsgrößen erhält, um daraus das Signal $\vec{y}_e(t+1)$ zu generieren. Dieses erweiterte Referenzsignal $\vec{y}_e(t+1)$ wird dann mit der Regelgröße $\vec{y}(t+1)$ verglichen, um einen modifizierten Regelfehler $\Delta \vec{y}_e(t+1)$ zu bestimmen, der anschließend zur Rückpropagation durch den neuronalen Beobachter genutzt wird und so zu einem veränderten Lernsignal $\Delta \vec{u}(t)$ führt. Die Berechnung des erweiterten Referenzsignals erfolgt anhand Gleichung 5.20:

$$\vec{y}_e(t+1) = \vec{y}_r(t+1) - \mathbf{A}(\vec{y}_r(t) - \vec{y}(t)) - \mathbf{B}(\vec{y}_r(t-1) - \vec{y}(t-1)) , \quad (5.20)$$

wobei für die Berechnung des Regelfehlers $\Delta \vec{y}(t+1)$ gilt, dass

$$\Delta \vec{y}(t+1) = \vec{y}_r(t+1) - \vec{y}(t+1) . \quad (5.21)$$

Für die Bestimmung des modifizierten Regelfehlers ergibt sich demnach:

$$\Delta \vec{y}_e(t+1) = \vec{y}_e(t+1) - \vec{y}(t+1) . \quad (5.22)$$

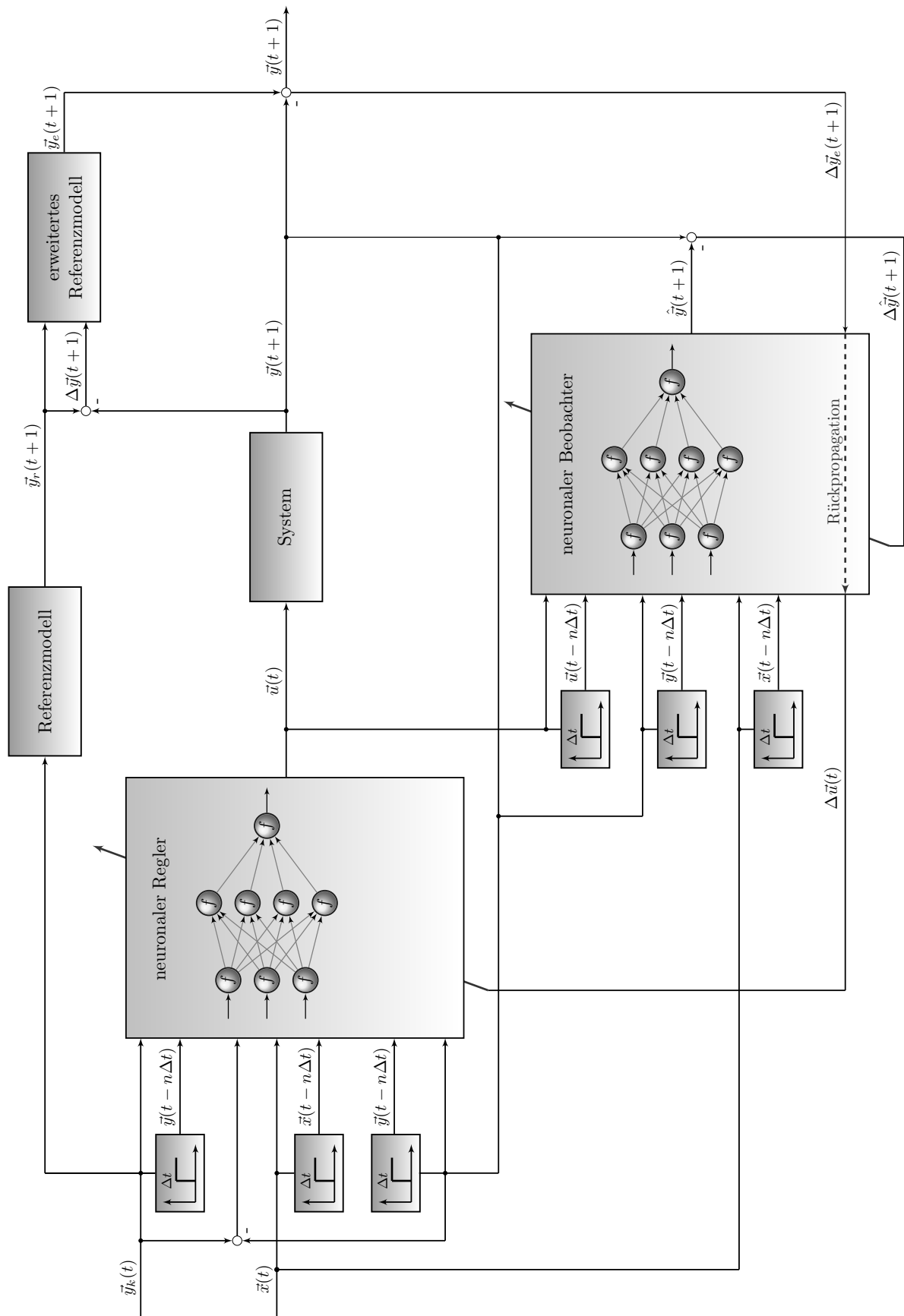


Abbildung 5.7: Darstellung des erweiterten modellfreien neuronalen Flugregelkreises.

Es sei angemerkt, dass **A** und **B** aus Gleichung 5.20 Hurwitz-Matrizen sind und demzufolge stabil. Fasst man die Gleichungen 5.20 und 5.21 zusammen, kann man für die Ausgabe des erweiterten Referenzmodells auch folgenden Zusammenhang finden:

$$\vec{y}_e(t+1) = \vec{y}_r(t+1) - \mathbf{A} \cdot \Delta\vec{y}(t) - \mathbf{B} \cdot \Delta\vec{y}(t-1). \quad (5.23)$$

Aus Gleichung 5.23 wird deutlich, dass die Erweiterung des Referenzmodells darin besteht, dass die Regelabweichung $\Delta\vec{y}$ der vergangenen zwei Zeitschritte t und $t-1$ über die Wichtungen **A** und **B** in die Berechnung von $\vec{y}_e(t+1)$ einfließt. Die genaue Verschaltung der einzelnen Signale ist anhand von Abbildung 5.8 nachzuvollziehen.

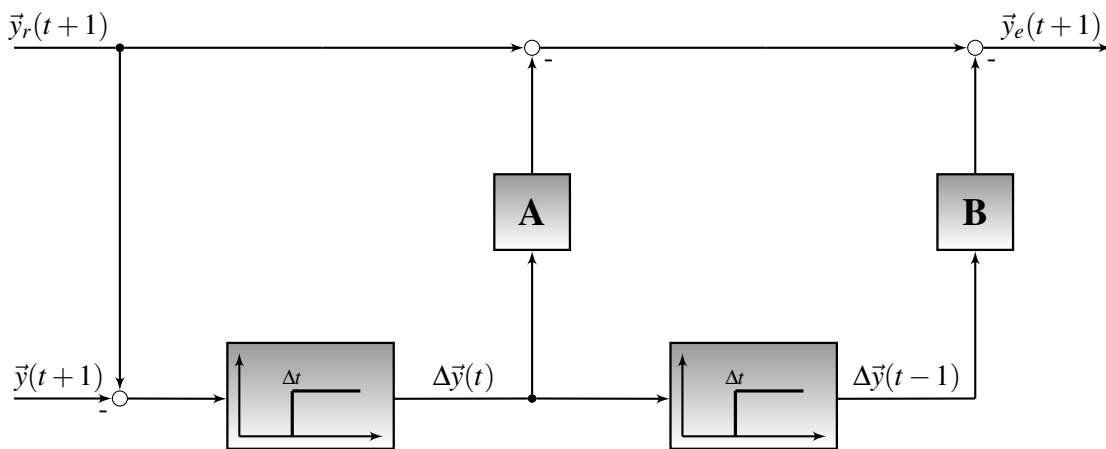


Abbildung 5.8: Darstellung des erweiterten Referenzmodells.

Dadurch, dass vergangene Fehlersignale in die Berechnung der erweiterten Referenzgröße einbezogen werden, erfolgt eine zusätzliche Glättung dieses Signals, da allzu große Abweichungen zwischen Referenzmodell und realem System reduziert werden. Die Dynamik des Referenzmodells wird in gewisser Weise durch die tatsächlich messbare Fehlerdynamik, die aus den vergangenen Regelfehlern $\Delta\vec{y}$ abgeleitet werden kann, angepasst. Dadurch wird letztlich das abstrahierte dynamische Verhalten des Referenzmodells mit dem realen Folgeverhalten des Systems kombiniert und erweitert. Damit wird auch verhindert, dass eventuelle sprunghafte Änderungen in der Regelabweichung $\Delta\vec{y}(t+1)$ gänzlich auf die Generierung des Lernsignals $\Delta\vec{u}(t)$ wirken und damit eine überproportionale, möglicherweise destabilisierend wirkende Gewichtsänderung im neuronalen Regler stattfindet. Hinsichtlich der Parameterwahl wird in [17] vorgeschlagen, die Matrizen **A** und **B** im Verhältnis $\mathbf{B} \approx 0,1 \cdot \mathbf{A}$ zu wählen. Diese Annahme ist für unbemannte Kleinflugzeuge in umfangreichen Untersuchungen der Parameter des erweiterten Referenzmodells in [82] nur zum Teil bestätigt worden, da sich auch höhere Werte für **B** als sinnvoll erwiesen haben.

5.3.1 Stabilitätsbetrachtung nach Ljapunow

Bereits in Abschnitt 5.1 ist auf die hohe Bedeutung eines stabilen Trainingssignals für jeden lernfähigen Regelkreis hingewiesen worden. Da nun die Ausgabe des erweiterten Referenzmo-

dells $\vec{y}_e(t+1)$ dafür genutzt wird, die modifizierte Regelabweichung $\Delta\vec{y}_e(t+1)$ zu bestimmen, aus der das Lernsignal des neuronalen Reglers ermittelt wird, soll in Anlehnung an [17] im Folgenden eine Stabilitätsbetrachtung des vorgestellten Ansatzes erfolgen. Dafür muss, wie bereits in Kapitel 3 für zeitdiskrete Systeme dargestellt, für die Stabilitätsfunktion folgendes gelten:

$$\dot{V} \triangleq \Delta V < 0 . \quad (5.24)$$

Eine geeignete Ljapunow-Funktion kann über den Regelfehler definiert werden:

$$V(\Delta\vec{y}(t)) = \Delta\vec{y}^T(t) \cdot \Delta\vec{y}(t) . \quad (5.25)$$

Berücksichtigt man nun, dass für die zeitliche Änderung der Ljapunow-Funktionen der Zusammenhang $\Delta V = V(t+1) - V(t)$ genutzt werden kann, lässt sich mit Gleichung 5.25 schreiben, dass

$$\Delta V(\Delta\vec{y}(t)) = V(\Delta\vec{y}(t+1)) - V(\Delta\vec{y}(t)) = \Delta\vec{y}^2(t+1) - \Delta\vec{y}^2(t) . \quad (5.26)$$

Setzt man weiterhin den Zusammenhang aus Gleichung 5.22 in Gleichung 5.23 ein, so ergibt sich nach Umstellung:

$$\Delta\vec{y}(t+1) = (\vec{y}_e(t+1) - \vec{y}(t+1)) + \mathbf{A} \cdot \Delta\vec{y}(t) + \mathbf{B} \cdot \Delta\vec{y}(t-1) , \quad (5.27)$$

womit der Ausdruck $\Delta\vec{y}(t+1)$ bestimmt ist. Dieser wird nun in Gleichung 5.26 eingesetzt, so dass die zeitliche Änderung der Ljapunow-Funktion gemäß Gleichung 5.28 umformuliert werden kann.

$$\Delta V(\Delta\vec{y}(t)) = [\Delta\vec{y}_e(t+1) + \mathbf{A} \cdot \Delta\vec{y}(t) + \mathbf{B} \cdot \Delta\vec{y}(t-1)]^2 - \Delta\vec{y}^2(t) . \quad (5.28)$$

Durch Ausmultiplizieren erhält man weiterhin:

$$\begin{aligned} \Delta V(\Delta\vec{y}(t)) &= \Delta\vec{y}_e^2(t+1) + \Delta\vec{y}_e^T(t+1) \cdot \mathbf{A} \cdot \Delta\vec{y}(t) + \Delta\vec{y}_e^T(t+1) \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) \\ &\quad + \Delta\vec{y}^T(t) \cdot \mathbf{A}^T \cdot \Delta\vec{y}_e(t+1) + \Delta\vec{y}^2(t) \cdot \mathbf{A}^2 \\ &\quad + \Delta\vec{y}^T(t) \cdot \mathbf{A}^T \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) + \Delta\vec{y}^T(t-1) \cdot \mathbf{B}^T \cdot \Delta\vec{y}_e(t+1) \\ &\quad + \Delta\vec{y}^T(t-1) \cdot \mathbf{B}^T \cdot \mathbf{A} \cdot \Delta\vec{y}(t) + \Delta\vec{y}^2(t-1) \cdot \mathbf{B}^2 - \Delta\vec{y}^2(t) . \end{aligned} \quad (5.29)$$

Formt man dann Gleichung 5.29 um, folgt schließlich folgender Zusammenhang:

$$\begin{aligned} \Delta V(\Delta\vec{y}(t)) &= \Delta\vec{y}^T(t) \cdot (\mathbf{A}^2 - \mathbf{E}) \cdot \Delta\vec{y}(t) + 2 \cdot \Delta\vec{y}_e^T(t+1) \cdot \mathbf{A} \cdot \Delta\vec{y}(t) \\ &\quad + 2 \cdot \Delta\vec{y}_e^T(t+1) \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) \\ &\quad + 2 \cdot \Delta\vec{y}^T(t) \cdot \mathbf{A}^T \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) \\ &\quad + \Delta\vec{y}^T(t-1) \cdot \mathbf{B}^T \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) + \Delta\vec{y}_e^2(t+1) . \end{aligned} \quad (5.30)$$

Führt man die Bedingung $\mathbf{Q} = -(\mathbf{A}^2 - \mathbf{E})$ ein, so kann sichergestellt werden, dass für die Ableitung der Ljapunow-Funktion $\Delta V(\Delta\vec{y}(t)) < 0$ gilt, wenn:

$$\begin{aligned} \Delta\vec{y}^T(t) \cdot \mathbf{Q} \cdot \Delta\vec{y}(t) &> 2 \cdot \Delta\vec{y}_e^T(t+1) \cdot \mathbf{A} \cdot \Delta\vec{y}(t) + 2 \cdot \Delta\vec{y}_e^T(t+1) \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) \\ &\quad + 2 \cdot \Delta\vec{y}^T(t) \cdot \mathbf{A}^T \cdot \mathbf{B} \cdot \Delta\vec{y}(t-1) + \Delta\vec{y}^2(t-1) \cdot \mathbf{B}^2 + \Delta\vec{y}_e^2(t+1) . \end{aligned} \quad (5.31)$$

Daraus ist ersichtlich, dass $\Delta \vec{y}(t)$ gegen Null geht, wenn maßgeblich der modifizierte Regelfehler $\Delta \vec{y}_e(t+1)$ die Ungleichung 5.31 erfüllt. Durch die Wahl von $0 < \mathbf{A} < 1$ und $\mathbf{B} < \mathbf{A}$ kann gemäß Gleichung 5.27 sichergestellt werden, dass der modifizierte Regelfehler $\Delta \vec{y}_e$ schneller sinkt als der Regelfehler $\Delta \vec{y}$, wodurch Ungleichung 5.31 schließlich Gültigkeit besitzt [17].

5.4 Implementierung des Vorwissens

Wie bereits angedeutet und unter anderem auch in [63] angewendet, ist es für die Flugregelungsanwendung notwendig, ein adäquates Vorwissen in den neuronalen Regelungselementen zu implementieren. Dies steht im Gegensatz zu der in [17] getroffenen Aussage, dass Vorwissen für die modellfreie Regelung nicht notwendig sei. Für den Flugregelungsfall lässt sich dies mit der Tatsache begründen, dass im Betrieb keine Lernphase nur minderer Regelungsgüte zu akzeptieren ist, da nicht zuletzt beim Start unzureichende Steuerkommandos schnell zu einem kritischen Flugzustand oder gar zum Absturz führen können. Daher ist es erforderlich, den zu implementierenden neuronalen Netzen gemäß der in den Abschnitten 2.6 und 2.7 getroffenen Aussagen ein fundiertes Vorwissen in einem reproduzierbaren Lernprozess anzutrainieren. Dazu soll der in Abbildung 2.6 dargestellte Entwurfsprozess herangezogen werden.

Da die Definition der Ein- und Ausgangssignale sowohl für den neuronalen Regler als auch für den Beobachter bereits in Abschnitt 5.2 erfolgt ist, stellt die Zusammenstellung der Trainingsmuster den nächsten Prozessschritt dar. Für das Anwendungsbeispiel eines unbemannten Fluggerätes stellt sich die Frage, wie die Trainingsmuster gewonnen werden sollen und welche flugdynamischen Bereiche diese abdecken. Zur Generierung der Trainingsdaten soll die in Kapitel 4 beschriebene Simulationsumgebung genutzt werden, wobei die Flugsteuerung von dem in [113] vorgestellten linearen Kaskadenregler übernommen wird. Dieser nach den in [28] beschriebenen Entwurfsmethoden ausgelegte Regelkreis liefert also mit seinen Eigenschaften das Basiswissen, das es im weiteren Betrieb zu erweitern gilt. Die dazu verwendete Flugbahn soll zum einen über eine ausreichende zeitliche Länge verfügen und zum anderen möglichst viele Manöver der Flugenveloppe eines unbemannten Kleinflugzeuges enthalten. Die unter diesen Aspekten gewählte Referenzflugbahn ist in Abbildung 5.9 qualitativ dargestellt. Sie weist eine Länge von 180 Sekunden auf und beinhaltet neben dem Reiseflug auch hochdynamische Manöver wie scharfe Rechts- und Linkskurven, die auch in Steig- und Sinkflügen durchgeführt werden. Neben der Forderung, ein möglichst breites Spektrum an Flugzuständen darzustellen, stellt diese Bahn auch sicher, dass kein Zustand in den Trainingsmustern überrepräsentiert ist, da dies zu unerwünschten Verzerrungen im Lernprozess führen würde.

Die Aufzeichnung der Trainingsmuster erfolgt mit 50 Hz, so dass mit insgesamt 9000 Datensätzen eine ausreichende Menge zu Verfügung steht. Um ein adäquates Verhalten der Netzwerke auf äußere Störungen zu gewährleisten, werden die Trainingsdaten unter dem Einfluss von Wind und zusätzlicher Turbulenz aufgezeichnet. Als Lernverfahren wird der in Abschnitt 2.3.3 beschriebene Levenberg-Marquardt-Algorithmus verwendet, der die Gewichtsänderungen in ge-

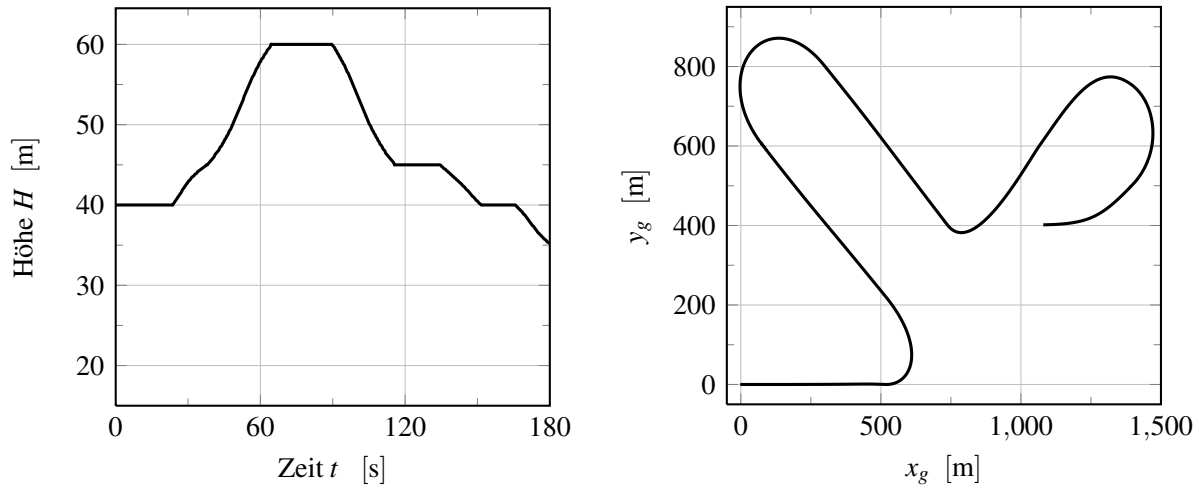


Abbildung 5.9: Profil der Referenzflugbahn in Längs- und Seitenbewegung.

bündelter Form durchführt (*batch learning*). Außerdem werden alle Muster normalisiert, damit die nichtlinearen Transferfunktionen nicht in ihre Sättigungsbereiche geraten. Um die Generalisierungseigenschaften sicherzustellen, wird das Konzept der Kreuzvalidierung aus Abschnitt 2.6.2 angewendet und mit einem Abbruchkriterium für vorzeitiges Trainingsende verknüpft. Um weiterhin die statistische Bewertung des Lernerfolges zu gewährleisten, werden 200 Netzwerke sowohl für das Regler- als auch für das Beobachternetzwerk trainiert. Diese sind in zehn Gruppen zu je 20 Netzen aufgeteilt, wobei in jeder Gruppe eine unterschiedliche Netztopologie trainiert wird, deren Startgewichte jeweils zufällig gewählt werden. Damit sind 200 verschiedene Startpunkte für den Lernprozess und seine Bewertung gegeben, wodurch ein lediglich zufälliger Lernerfolg einzelner Netzwerke unwahrscheinlich ist. Die Topologien haben in der Regel eine verdeckte Schicht, die in ihrer Neuronenzahl zwischen acht und 40 variiert; zu Vergleichszwecken verfügen zwei Topologien über zwei verdeckte Schichten mit fünf bis zehn Neuronen in den Schichten. Eine hierdurch signifikante Verbesserung der Netzwerkeigenschaften konnte jedoch nicht beobachtet werden. Alle Netze werden, vom vorzeitigen Abbruch abgesehen, über 300 Epochen trainiert, wobei dieser Wert empirisch ermittelt ist. Es ist auch eine deutlich höhere Zahl an Epochen denkbar, allerdings haben Untersuchungen des vorliegenden Lernproblems gezeigt, dass die sich ergebenden Reduzierungen der Ausgabefehler marginal sind und das Training häufig durch die Kreuzvalidierung vorzeitig abgebrochen wird. Als Bewertungsmaß des Lernerfolges wird der dafür üblich genutzte mittlere quadratische Fehler herangezogen und für alle Netzwerke ein niedriger Zielwert von 10^{-6} festgelegt. Bei Berechnung des MSE wird der quadratische Fehler aus Gleichung 2.10 über alle Trainingsmuster gemittelt. Neben anderen Kriterien ist als Gütemaß ebenfalls die Wurzel des MSE denkbar, da dieser die Bewertung der Lernergebnisse in der ursprünglichen Dimension der Trainingsmuster erlaubt. Die Betrachtung der Fehler erfolgt jeweils getrennt für Trainings-, Test- und Validierungsdaten, um sowohl eine Aussage über den eigentlichen Trainingsverlauf, als auch über die zu erwartende Robustheit der Netzantwort bei unbekannten Eingangssignalen zu bewerten.

5.4.1 Bewertung des statistischen Trainings

Die statistische Auswertung des Entwurfs- und Trainingsprozesses soll im Folgenden am Beispiel der 200 Reglernetzwerke vorgenommen werden, da deren implementiertes Vorwissen gerade zu Beginn eines Fluges großen Einfluss auf die Regelgüte hat. Die Ergebnisse des Beobachtertrainings sind generell von gleicher Qualität [103] und werden daher nur bei der konkreten Netzauswahl herangezogen. Im ersten Schritt zur Bewertung des Lernerfolges ist es zweckmäßig, ein geeignetes statistisches Gütemaß zu wählen, das es ermöglicht, alle Netzwerke eines betrachteten Trainingsproblems ohne Differenzierung ihrer Topologie auszuwerten. Dadurch kann sehr schnell eine allgemeine Aussage darüber getroffen werden, ob alle Netzwerke eine hinreichend gute Abbildung erzeugen oder ob nur wenigen dies eher zufällig gelingt. Eine Möglichkeit hierzu ist die Bildung einer Verteilungsfunktion des mittleren quadratischen Fehlers aller trainierten Netzwerke. Diese gibt Auskunft über die Wahrscheinlichkeit, mit der ein Wert für den quadratischen Fehler durchschnittlich erreicht wird. Für die trainierten Reglernetzwerke ist dieser nun in Abbildung 5.10 in einem typischen Verlauf dargestellt. Bezogen auf die vorigen Erläuterungen enthält diese die statistische Auswertung von 200 Netzwerken, die über 300 Epochen mit jeweils 9000 Trainingsmustern trainiert wurden. Gleichzeitig wird die zufällige Auswahl der 70 % Trainingsdaten sowie der jeweils 15 % Test- und Validierungsdaten in jeder Epoche erneut durchgeführt, womit eine aussagekräftige Datenbasis gegeben ist.

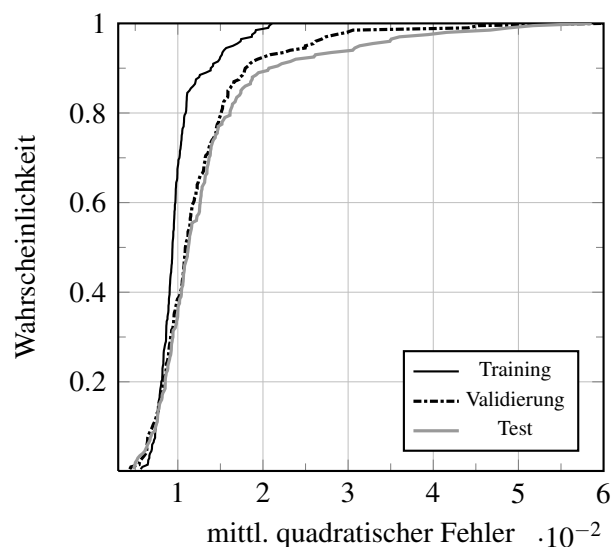


Abbildung 5.10: Verteilungsfunktion des Fehlers aller Reglernetzwerke.

Zuvorderst ist an der Kurve der Testdaten erkennbar, dass der MSE bezogen auf alle Netzwerke und alle Datensätze stets geringer als $5,85 \cdot 10^{-2}$ ausfällt. Betrachtet man den dazugehörigen RMSE, so lässt sich feststellen, dass der Fehler des erlernten Querruderausschlages aller genutzten Daten im ungünstigsten Fall $0,24^\circ$ beträgt. Dies ist ein geringer Wert, bedenkt man, dass die Trainingsmuster den störenden Einfluss von Wind und Turbulenz enthalten. Darüber hinaus ist ersichtlich, dass der MSE für 90 % aller Fälle bereits geringer als $2,0 \cdot 10^{-2}$ ist, was

nahezu einer Halbierung des RMSE auf $0,14^\circ$ entspricht. Dies ist besonders für die Test- und Validierungsfehler zu betonen, da diese ein Maß für die Robustheit und die Generalisierungsfähigkeit der Netzwerke darstellen. Die Kurven von Trainings-, Test- und Validierungsdaten nähern sich mit steigender Abbildungsgenauigkeit deutlich an, so dass alle Verläufe bei einer Wahrscheinlichkeit von 0,4 ungefähr gleichmäßig bei einem MSE von $1,04 \cdot 10^{-2}$ liegen und sich der geringste MSE schließlich bei $4,85 \cdot 10^{-3}$ einstellt. Dieser Gesamtverlauf zeigt in schlüssiger Weise, dass dieses komplexe Lernproblem durchgehend angemessen und bei hoher Robustheit von den trainierten Netzen abgebildet werden kann.

Obgleich Abbildung 5.10 gute Gesamtergebnisse erkennen lässt, existieren bei Test- und Validierungsdaten Unterschiede im MSE von ca. $5,0 \cdot 10^{-2}$ über alle Netzwerke. In Anlehnung an Abschnitt 2.6.2 ist daher die Identifizierung einer für das Abbildungsproblem möglichst optimalen Topologie der nächste Schritt. Zu diesem Zweck wird der Fehler in Abhängigkeit der Netzwerkgröße betrachtet, wobei der MSE der 20 Netzwerke einer Topologie für Trainings-, Test- und Validierungsdaten zu einem gemittelten Wert zusammengefasst ist. Daraus ergeben sich Kurven, anhand derer der MSE bezogen auf die Anzahl der verdeckten Neuronen, respektive der Gewichte abgelesen werden kann. Diese Verläufe sind in Abbildung 5.11 dargestellt, wobei angemerkt sei, dass die beiden vierschichtigen Topologien auf der Abszisse separat gekennzeichnet sind und die Sortierung nach aufsteigender Zahl der Gewichte erfolgt.

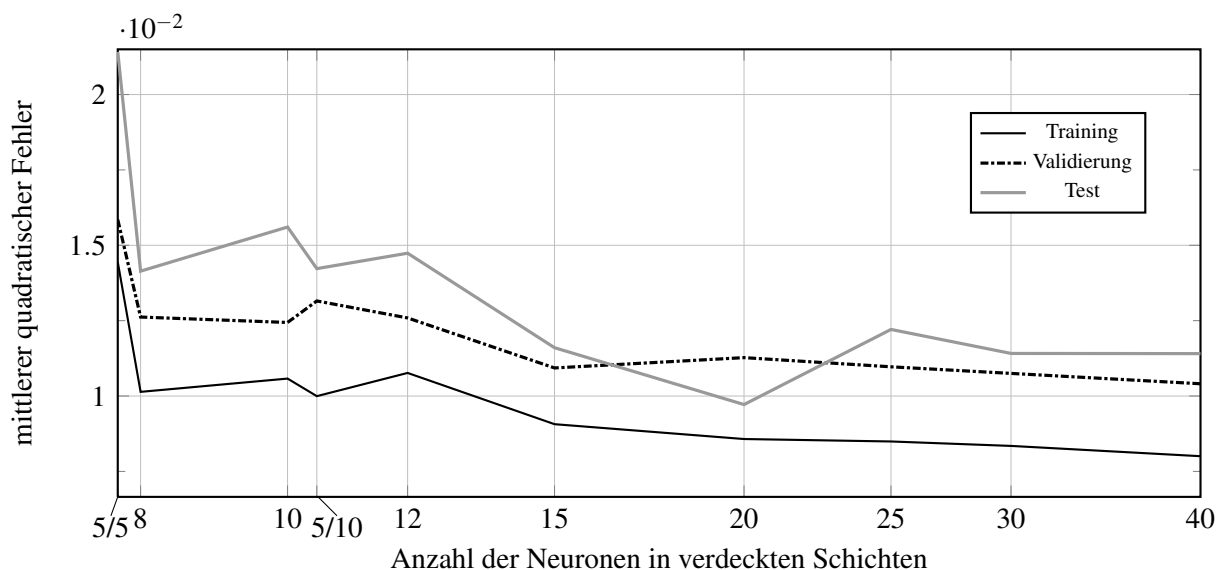


Abbildung 5.11: Verlauf des Fehlers über der Anzahl verdeckter Neuronen aller Reglernetze.

Es wird deutlich, dass diese Verläufe generell jenen ähneln, die bereits in Abbildung 2.5 für einzelne Netze skizziert sind. Abbildung 5.11 stellt damit im Grunde eine Erweiterung der graphischen Darstellung der Generalisierungsfähigkeit für mehrere Netze unterschiedlicher Größe dar. Wie zuvor erläutert, existiert eine bezogen auf die Komplexität des Lernproblems optimale Netzwerkgröße, die aus dem Validierungs- und Testfehler abgeleitet werden kann: Gibt es

zu viele Gewichte im Netz, werden die Trainingsmuster ohne Verallgemeinerung auswendig gelernt, bei zu wenigen sinkt die Abbildungsgenauigkeit. Abbildung 5.11 zeigt, dass ein Minimum der relevanten Test- und Validierungsfehler bei den Netzwerken mit 15 und 20 vorliegt, wobei der Validierungsfehler bei beiden nahezu gleich ist. Für das vorliegende Lernproblem scheinen Netze, die deutlich weniger als 15 Neuronen in der verdeckten Schicht aufweisen, im Sinne der Genauigkeit weniger gut geeignet zu sein; auch eine weitere Schicht kann dies nicht nachhaltig ändern. Gleichzeitig steigt bei größeren Netzwerken der Testfehler, während der Validierungsfehler ungefähr gleich bleibt. Damit ist von einem größeren und damit in der Berechnung aufwändigeren Netzwerk kein Gewinn in der Abbildungsgenauigkeit bei gleichzeitiger Robustheit zu erwarten. Daher kann die Netzauswahl durch statistische Auswertung des Trainings auf die Netzwerke mit 15 und 20 verdeckten Neuronen eingegrenzt werden.

5.4.2 Bewertung des Reglertrainings

Die erläuterte systematische Auswertung aller trainierten Netzwerke ermöglicht die Identifizierung der Topologien möglichst hoher Güte. Für den neuronalen Regler ist auf dieser Basis ein Netzwerk mit 20 Neuronen in der verdeckten Schicht ausgewählt worden. Die Wahl ist damit zu begründen, dass diese Gruppe zum einen einen geringeren Testfehler aufweist und zum anderen die höhere Gewichteanzahl das robuste Hinzulernen im Betrieb eher gewährleistet als ein kleineres Netz mit zu wenigen Freiheitsgraden. In Abbildung 5.12 sind nun für die Referenzbahn die Zielwerte des Querrudersignals und die dazugehörigen Antworten der Trainingsdaten dargestellt, wobei dieser Verlauf als exemplarisch für die durchgehend guten Lernergebnisse aller trainierten Netzwerke bezeichnet werden kann.

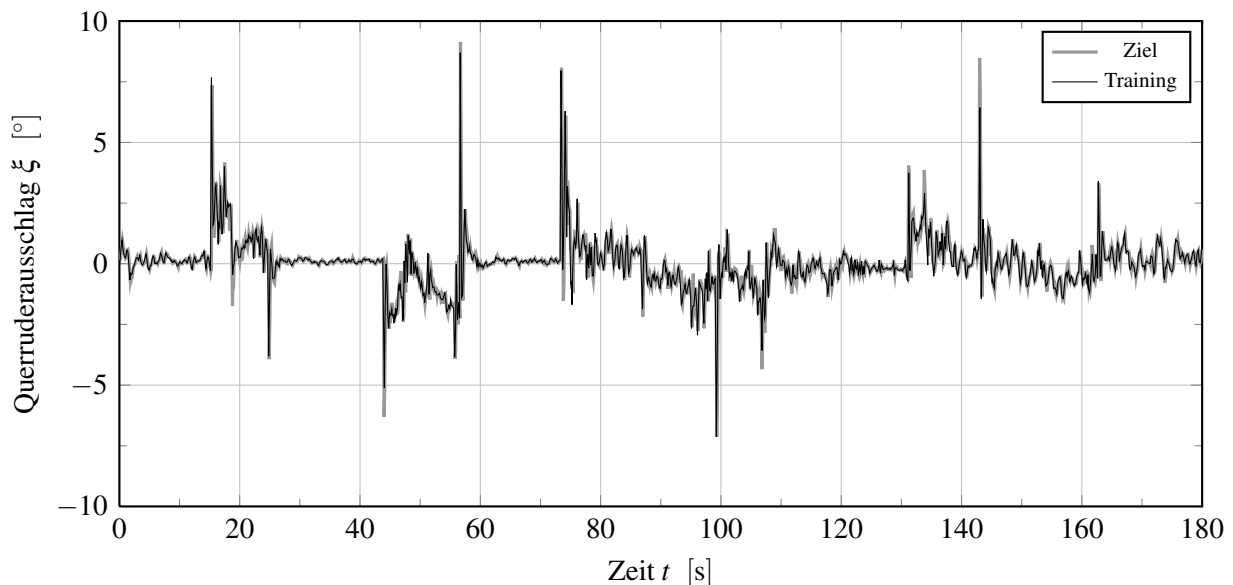


Abbildung 5.12: Antwortverhalten eines Reglernetzwerkes für die Referenzflugbahn.

Sehr gut ist die große Übereinstimmung der beiden Zeitverläufe zu erkennen, was sich nicht zuletzt in einem geringen mittleren quadratischen Fehler von $7,61 \cdot 10^{-3}$ widerspiegelt. Festzustellen ist auch, dass die wenigen starken Ruderkommandos, die beim Einleiten scharfer Kurven auftreten, ebenfalls gut abgebildet werden. Zusätzlich ist ersichtlich, dass der Querruderverlauf generell ein gewisses Rauschverhalten aufweist, was der Ausregelung der Turbulenzeinflüsse zuzuschreiben ist. Man kann zusammenfassend feststellen, dass das betrachtete Lernproblem augenscheinlich in hoher Qualität von neuronalen Netzwerken abgebildet wird. In obiger Abbildung ist im Sinne der Deutlichkeit bewusst auf die Darstellung der Test- und Validierungsantworten verzichtet worden. Diese sind aber besonders interessant, um die Generalisierungsfähigkeit und die Robustheit eines KNN zu bewerten. Aus diesem Grund sind die Ergebnisse von Trainings-, Test- und Validierungsdaten in Abbildung 5.13 in einem Streudiagramm (engl. *scatter plot*) dargestellt. Bei dieser Art der Darstellung werden die tatsächlichen Ausgaben über den Zielwerten abgebildet, so dass sich im Idealfall eine Gerade mit der Steigung von eins ergibt und den Erfolgsgrad des Lernprozesses sehr leicht interpretierbar macht.

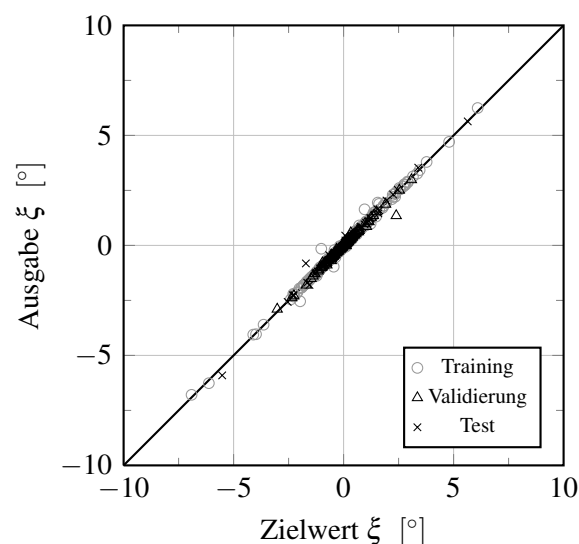


Abbildung 5.13: Streudiagramm von Zielwerten und Netzantworten eines Reglernetzwerkes.

Da die Darstellung aller 9000 Datenpunkte für eine solche Abbildung nicht zielführend ist, sind 10 % der Trainingsmuster zufällig ausgewählt worden. Es ist gut zu erkennen, dass sich die Punkte sowohl für Training als auch für Test und Validierung bis auf geringe Abweichungen sehr nah der Ideallinie annähern. Dies zeigt sich auch in geringen Fehlern für die Test- und Validierungsdaten, die in Tabelle 5.2 für das betrachtete Netzwerk zusammengefasst sind. Diese geringen Fehler unterstreichen, dass das betrachtete Reglernetzwerk auch im Fall von Störungen, hervorgerufen durch das Turbulenzspektrum, robust reagiert und demnach über hinreichende Verallgemeinerungsfähigkeiten verfügt. Daraus lässt sich letztlich ableiten, dass das Basiswissen dieses analysierten Reglernetzwerkes wegen seiner Robustheit gut für die Anwendung im Regelkreis geeignet ist.

| | MSE | RMSE [°] |
|-------------|--------|----------|
| Training | 0,0076 | 0,0871 |
| Validierung | 0,0123 | 0,1109 |
| Test | 0,0091 | 0,0953 |

Tabelle 5.2: Fehlerbetrachtung des neuronalen Reglers.

5.4.3 Bewertung des Beobachtertrainings

Da das Beobachternetzwerk Bestandteil des Kritikelementes ist, das das Lernsignal für den neuronalen Regler liefert, muss auch für dieses der Trainingsverlauf des Basiswissens bewertet werden. Als Beispiel wird ein Netz mit 25 Neuronen in der verdeckten Schicht gewählt. Analog zum Reglernetzwerk ist in Abbildung 5.14 der Zeitverlauf der Netzausgabe des Beobachters dargestellt. Dabei ist auch am Verlauf der Querschleunigung a_y der störende Einfluss des Turbulenzspektrums erkennbar. Da der qualitative Verlauf des Signals die Darstellung zulässt, sind die Ergebnisse der Test- und Validierungsmuster ebenfalls eingezeichnet. Ähnlich wie zuvor sind 5 % der Datenpunkte zufällig ausgewählt und abgebildet worden.

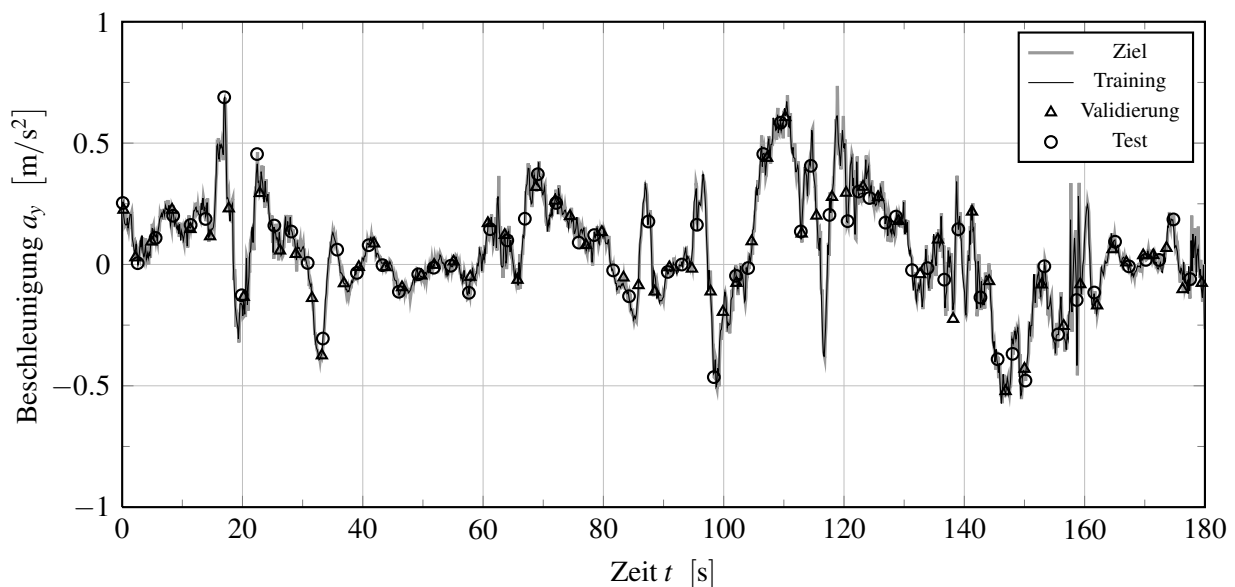


Abbildung 5.14: Antwortverhalten eines Reglernetzwerkes für die Referenzflugbahn.

Wie auch beim Reglernetzwerk wird eine gute Übereinstimmung zwischen Zielwerten und den Antworten aller Datensätze erzielt, was sich erneut in den Fehlerwerten niederschlägt. Diese sind in Tabelle 5.3 für den Beobachter zusammengestellt. Die erkennbar geringen Fehler für Validierungs- und Testdaten lassen ebenfalls auf gute Generalisierungs- und Robustheitseigenschaften schließen. Dass die Fehler generell deutlich geringer ausfallen als beim Reglertraining

ist zum einen darauf zurückzuführen, dass die Zielwerte des Beobachters in weiten Teilen um eine Größenordnung kleiner sind als die des Reglers. Zum anderen verfügt der Verlauf der Zielwerte des Beobachters nicht über so starke Spitzen wie der Querruderverlauf, was das Training stark begünstigt.

| | MSE | RMSE [m/s ²] |
|-------------|-----------------------|--------------------------|
| Training | $3,506 \cdot 10^{-7}$ | $5,921 \cdot 10^{-4}$ |
| Validierung | $3,480 \cdot 10^{-7}$ | $5,899 \cdot 10^{-4}$ |
| Test | $3,505 \cdot 10^{-7}$ | $5,920 \cdot 10^{-4}$ |

Tabelle 5.3: Fehlerbetrachtung des neuronalen Beobachters.

Zusammenfassend lässt sich feststellen, dass das gestellte Lernproblem gut von neuronalen Netzwerken abgebildet werden kann. Sowohl für den Regler, als auch für den Beobachter werden sehr gute Trainingsergebnisse erreicht, die belegen, dass das nötige Basiswissen robust erlernt werden kann. Die statistische Bewertung des Lernerfolges großer Gruppen von Netzwerken ermöglicht es dabei, eventuelle Zufälligkeiten zu minimieren. Die Gesamtergebnisse unterstreichen auch, dass die betrachteten Netzwerke geeignet sind, in einem lernfähigen Flugregelkreis zum Einsatz zu kommen.

6 Nichtlineare dynamische Inversion

Im Gegensatz zu dem in Kapitel 5 vorgestellten Ansatz der adaptiven Regelung stellt die dynamische Inversion ein nichtlineares Regelungsverfahren dar, das auf Kenntnis eines genauen physikalischen Modells basiert. Die Grundidee hierbei besteht darin, für ein nichtlineares System durch eine geeignete Zustandstransformation ein lineares arbeitspunktunabhängiges Eingangs-/Ausgangsverhalten zu realisieren [37, 38]. Ohne dass Vereinfachungen wie bei der klassischen Linearisierung getroffen zu werden brauchen, können dennoch lineare Regelungselemente für das resultierende Ersatzsystem verwendet werden. Dies wird durch eine linearisierende Zustandsrückführung erreicht, weshalb das Verfahren auch exakte Eingangs-/Ausgangslinearisierung genannt wird (engl. *feedback linearisation*). Letztlich ist es möglich, durch Einführung einer Ersatzregelgröße \vec{v} (engl. *pseudo control*) das eigentliche Systemverhalten zu invertieren, was in Abbildung 6.1 an einem stark abstrahierten Regelkreis erkennbar ist. Die Inversion des Streckenverhaltens kann folglich zur Bestimmung eines zum Einhalten einer kommandierten Führungsgröße nötigen Stellsignals genutzt werden. Der zusammengefasst vereinfacht als linearer Regler bezeichnete Block enthält seinerseits lineare Referenzmodelle und Fehlerregler.

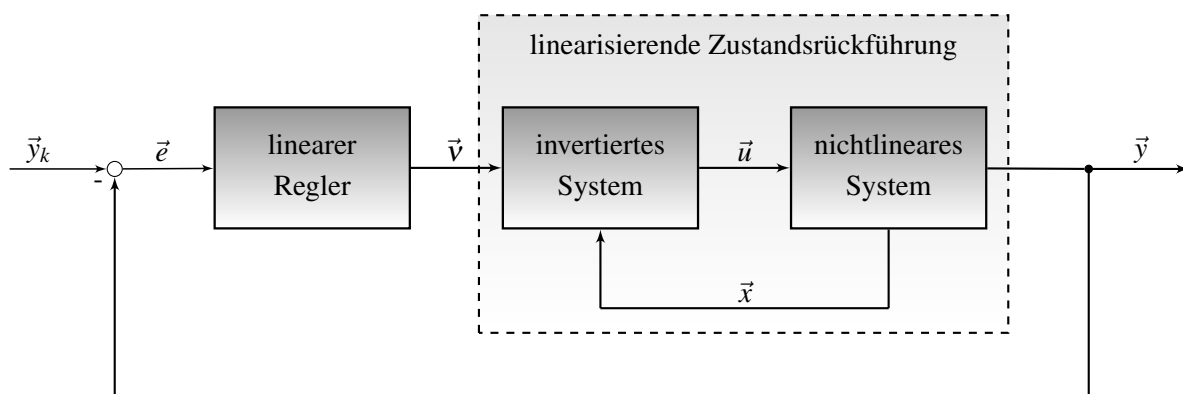


Abbildung 6.1: Vereinfachte Darstellung eines Regelkreises auf Basis dynamischer Inversion.

Das Konzept der dynamischen Inversion wurde seit Anfang der 1990er Jahre besonders in den USA systematisch für die regelungstechnische Anwendung in der Luft- und Raumfahrt untersucht [40–42, 44, 116]. Dabei wurde der Einsatz sowohl für Flächenflugzeuge als auch für Drehflügler simulativ und in praktischen Flugversuchen analysiert. Ein Grundproblem der dynamischen Inversion besteht darin, dass dieses Verfahren vergleichsweise empfindlich ge-

genüber Modellunsicherheiten reagiert, da bei ungenauer Systemidentifizierung das invertierte Streckenmodell stark vom realen System abweicht. Eine Möglichkeit, die daraus resultierenden Fehler zu kompensieren, besteht in ihrer Approximation durch neuronale Netzwerke, was zunächst im Feld der Robotik demonstriert wurde [39, 117] und anschließend auch in einer Fülle von Luftfahrtanwendungen dokumentiert ist [17, 43, 45, 47, 49, 52]. Um darüber hinaus dem Problem der Stellraten- und Ausschlagsbegrenzung von Aktuatoren zu begegnen, ist eine dynamische Beschränkung der Ersatzregelgröße möglich (engl. *pseudo control hedging*), wodurch destabilisierender Einfluss der limitierten Aktuatordynamik vermieden werden kann [43].

Weiterhin ist eine stetig steigende Zahl von Anwendungen für unbemannte Flugsysteme zu verzeichnen [52–55, 81, 118]. Diese Fluggeräte bieten sich in besonderem Maße für adaptive Regelungsstrategien an, da meist eine nur begrenzte Parameteridentifizierung erfolgt und lineare Regler daher eher konservativ ausgelegt werden müssen. Außerdem sind kleine Flugzeuge anfällig gegenüber starkem Wind und Turbulenz, was durch ein nichtlineares arbeitspunktunabhängiges Regelungssystem reduziert werden kann. Ein weiterer Vorteil des mit KNN kombinierten Ansatzes ist die Toleranz gegenüber Aktuatorausfällen und strukturellen Schäden am Flugzeug [51], da die Lernfähigkeit der neuronalen Netze eine Anpassung der Reglereigenschaften an Systemfehler erlaubt, bei denen lineare Ansätze instabil würden. Um dieses fehlertolerante Verhalten weiter auszubauen, ist es ebenfalls möglich, den Regelkreis mit einem lernfähigen Fehlerdetektionssystem zu kombinieren, was eine gezielte Rekonfiguration der Stellorgane im Falle eines Aktuator Schadens ermöglicht [50, 83]. Diese positiven Eigenschaften haben auch dazu beigetragen, dass derzeit systematische Analysen zur Verifikation und Validation im Sinne der Zertifizierbarkeit dieser neuronal gestützten Regler erfolgen [17, 35].

6.1 Grundprinzip der dynamischen Inversion

Im Folgenden werden die wichtigsten Aspekte der Regelung mittels dynamischer Inversion aufgezeigt; weiterführende Ausarbeitungen sind für den allgemeinen Fall in [37] und mit Schwerpunkt der Luftfahrtanwendung in [52] zu finden. Betrachtet werden nichtlineare Mehrgrößensysteme (engl. *multiple input, multiple output - MIMO*), die eine gleiche Anzahl von Eingangs- und Ausgangsgrößen aufweisen und linear in ihrem Steuerverhalten sind:

$$\begin{aligned}\dot{\vec{x}} &= \vec{f}(\vec{x}) + \mathbf{G}(\vec{x}) \cdot \vec{u} , \\ \vec{y} &= \vec{h}(\vec{x}) .\end{aligned}\tag{6.1}$$

Hierbei sind $\vec{x}_{n \times 1} = [x_1, x_2, \dots, x_n]^T$ der Zustandsvektor der Länge n , $\vec{u}_{m \times 1} = [u_1, u_2, \dots, u_m]^T$ der Eingangsvektor der Länge m und $\vec{y}_{m \times 1} = [y_1, y_2, \dots, y_m]^T$ der Ausgangsvektor des Systems. Die Matrix $\mathbf{G}_{n \times m}(\vec{x})$ besteht aus m Vektorfeldern $\vec{g}_{n \times 1}$ der Länge n , die sich genau wie die Vektorfelder \vec{f} und \vec{h} aus nichtlinearen Funktionen des Zustandsvektors \vec{x} zusammensetzen und mehrfach stetig differenzierbare Abbildungen darstellen.

Ziel der dynamischen Inversion ist es, durch nichtlineare Zustandstransformation eine lineare Ersatzdynamik zu realisieren und gemäß Abbildung 6.1 zur Bestimmung der Stellgrößen schließlich die Systemdynamik zu invertieren. Dazu ist Kenntnis über den Einfluss der Eingangsgrößen \vec{u} auf die Ausgangsgrößen \vec{y} zwingend erforderlich. Werden die Ausgaben nicht direkt von den Eingängen beeinflusst, sondern ihre r -ten Ableitungen, spricht man von einem relativen Grad des Systems. Diesen näher zu erläutern, soll im Sinne der Übersichtlichkeit zunächst an einem Eingrößensystem (engl. *single input, single output - SISO*) erfolgen, wie es in Gleichung 6.2 gegeben ist.

$$\begin{aligned}\dot{\vec{x}} &= \vec{f}(\vec{x}) + \vec{g}(\vec{x}) \cdot u \\ y &= h(\vec{x})\end{aligned}\tag{6.2}$$

Um nun die Ableitung der Ausgangsgröße \dot{y} bestimmen zu können, werden die sogenannten Lie-Ableitungen eingeführt, deren Rechenvorschriften in [37] im Detail erläutert sind:

$$L_f \cdot h(\vec{x}) = \frac{\partial h}{\partial \vec{x}} \cdot \vec{f}(\vec{x}) .\tag{6.3}$$

Dies entspricht der Ableitung der Funktion $h(\vec{x})$ entlang des Vektorfeldes $\vec{f}(\vec{x})$ und kann auch mehrfach durchgeführt werden, so dass für die k -te Ableitung gilt:

$$L_f^k \cdot h(\vec{x}) = L_f(L_f^{k-1} \cdot h(\vec{x})) .\tag{6.4}$$

Mit dieser Information kann unter Zuhilfenahme von Gleichung 6.3 nun die Ableitung der Ausgangsgröße \dot{y} gefunden werden, die vom Eingang u beeinflusst wird, was in Gleichung 6.5 dargestellt ist.

$$\dot{y} = \frac{\partial y}{\partial x} \cdot \frac{\partial x}{\partial t} = \frac{\partial h}{\partial \vec{x}} \cdot \dot{\vec{x}} = \frac{\partial h}{\partial \vec{x}} \cdot [\vec{f}(\vec{x}) + \vec{g}(\vec{x}) \cdot u] = L_f \cdot h(\vec{x}) + L_g \cdot h(\vec{x}) \cdot u\tag{6.5}$$

In dem Falle, dass die Eingangsgröße u an einem betrachteten Punkt \vec{x}_0 keinen Einfluss auf die Ausgangsgröße y besäße, also $L_g \cdot h(\vec{x}) = 0$, vereinfachte sich Gleichung 6.5 zu $\dot{y} = L_f \cdot h(\vec{x})$ und eine weitere Ableitung gemäß Gleichung 6.6 wäre notwendig.

$$y^{(2)} = \frac{\partial(L_f \cdot h)}{\partial \vec{x}} \cdot [\vec{f}(\vec{x}) + \vec{g}(\vec{x}) \cdot u] = L_f^2 \cdot h(\vec{x}) + L_g \cdot L_f \cdot h(\vec{x}) \cdot u\tag{6.6}$$

Dies kann bis zur r -ten Ableitung wiederholt werden, so dass ein expliziter Zusammenhang zwischen Eingang u und $y^{(r)}$ zu ermitteln ist:

$$y^{(r)} = L_f^r \cdot h(\vec{x}) + L_g \cdot L_f^{r-1} \cdot h(\vec{x}) \cdot u .\tag{6.7}$$

Die r -te Ableitung der Regelgröße y ist demnach durch den Eingang u direkt steuerbar, weshalb r auch als relativer Grad eines Systems bezeichnet wird. Da die Regelgröße y durch Integration aus den r Ableitungen gewonnen wird, ist der relative Grad auch ein Maß für die Phasenverschiebung zwischen y und Stellgröße u . Mit den bisherigen Überlegungen lässt sich Gleichung 6.7 auf ein Mehrgrößensystem gemäß Gleichung 6.1 in folgender Form übertragen:

$$\underbrace{\begin{bmatrix} y_1^{r_1} \\ y_2^{r_2} \\ \vdots \\ y_m^{r_m} \end{bmatrix}}_{[y_i^{(r_i)}]_{m \times 1}} = \underbrace{\begin{bmatrix} L_f^{r_1} \cdot h_1(\vec{x}) \\ L_f^{r_2} \cdot h_2(\vec{x}) \\ \vdots \\ L_f^{r_m} \cdot h_m(\vec{x}) \end{bmatrix}}_{\mathbf{b}(\vec{x})} + \underbrace{\begin{bmatrix} L_{g1} \cdot L_f^{r_1-1} \cdot h_1(\vec{x}) & \cdots & L_{gm} \cdot L_f^{r_1-1} \cdot h_1(\vec{x}) \\ L_{g1} \cdot L_f^{r_2-1} \cdot h_2(\vec{x}) & \cdots & L_{gm} \cdot L_f^{r_2-1} \cdot h_2(\vec{x}) \\ \vdots & \ddots & \vdots \\ L_{g1} \cdot L_f^{r_m-1} \cdot h_m(\vec{x}) & \cdots & L_{gm} \cdot L_f^{r_m-1} \cdot h_m(\vec{x}) \end{bmatrix}}_{\mathbf{A}(\vec{x})} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}}_{\vec{u}} \quad (6.8)$$

Mit Einführung der sogenannten Entkopplungsmatrix $\mathbf{A}(\vec{x})$ kann nun in Anlehnung an [37, 52, 54] der relative Grad r eines Mehrgrößensystems definiert werden.

Ein nichtlineares Mehrgrößensystem gemäß Gleichung 6.1 besitzt an einem Punkt $\vec{x}_0 \in R^n$ einen relativen Grad $[r_1, r_2, \dots, r_m]$, wenn:

- $L_{g_j} \cdot L_f^k \cdot h_i(\vec{x}) = 0 \quad \forall 1 \leq j \leq m, k < r_i - 1, 1 \leq i \leq m$ und \vec{x} in einer Umgebung von \vec{x}_0
- und die Entkopplungsmatrix $\mathbf{A}_{m \times m}(\vec{x})$ bei $\vec{x} = \vec{x}_0$ invertierbar ist.

Der relative Grad eines Mehrgrößensystems berechnet sich nach Gleichung 6.9, wobei dieser stets kleiner ist als die Ordnung n des Systems.

$$r = r_1 + \dots + r_m = \sum_{i=1}^m r_i \leq n \quad (6.9)$$

Der relative Grad ist eine notwendige Bedingung, um eine geeignete Zustandstransformation zu finden, die eine linearisierende Zustandsrückführung erlaubt; außerdem ist bei Mehrgrößensystemen die Nichtsingularität der Entkopplungsmatrix eine wichtige Voraussetzung [37, 52].

6.1.1 Nichtlineare Zustandstransformation

Durch eine passende Koordinatentransformation der Zustandsgrößen \vec{x} kann erreicht werden, dass das transformierte System eine lineare Eingangs-/Ausgangsdynamik besitzt. In Anlehnung an [37, 38, 52] soll dies im Sinne der Übersichtlichkeit anhand eines Eingrößensystems erläutert werden, wie es in Gleichung 6.2 gegeben ist. Besitzt ein nichtlineares System einen relativen Grad r , werden die Lie-Ableitungen $L_g L_f^k h(\vec{x})$ zu Null, solange $k < r - 1$ ist. Damit kann die in Gleichung 6.10 dargestellte Transformation des Systems gefunden werden.

$$\begin{aligned} z_1 &= \xi_1 = \Phi_1(\vec{x}) = y = L_f^0 \cdot h(\vec{x}) = h(\vec{x}) \\ z_2 &= \xi_2 = \Phi_2(\vec{x}) = \frac{d\Phi_1}{dt} = \dot{z}_1 = \dot{y} = L_f^1 \cdot h(\vec{x}) = L_f \cdot h(\vec{x}) \\ &\vdots \\ z_{r-1} &= \xi_{r-1} = \Phi_{r-1}(\vec{x}) = \frac{d\Phi_{r-2}}{dt} = \dot{z}_{r-2} = y^{(r-2)} = L_f^{r-2} \cdot h(\vec{x}) \\ z_r &= \xi_r = \Phi_r(\vec{x}) = \frac{d\Phi_{r-1}}{dt} = \dot{z}_{r-1} = y^{(r-1)} = L_f^{r-1} \cdot h(\vec{x}) \end{aligned} \quad (6.10)$$

Hiermit wird deutlich, dass die ersten r Zustände des Systems transformiert werden können. Für die $n - r$ weiteren Zustände eines Systems n -ter Ordnung müssen ebenso Transformationen gefunden werden. Dabei ist sicherzustellen, dass die Abbildung $\vec{z} = \Phi(\vec{x})$ einen Diffeomorphismus darstellt [52, 62]. Das heißt, dass die Matrix $\partial\Phi(\vec{x})/\partial\vec{x}$ an der Stelle der Transformation $\vec{x} = \vec{x}_0$ invertierbar ist und eine Rücktransformation $\vec{x} = \Phi^{-1}(\vec{z})$ existiert. Dies bedingt nach [37], dass die Zeilenvektoren dieser Matrix linear unabhängig sein müssen, womit schließlich die zusätzlichen Zustandstransformationen gebildet werden können mit:

$$\begin{aligned} z_{r+1} &= \eta_1 = \Phi_{r+1}(\vec{x}) , \\ &\vdots \\ z_n &= \eta_{n-r} = \Phi_n(\vec{x}) . \end{aligned} \quad (6.11)$$

Unter Verknüpfung der Gleichungen 6.10 und 6.11 kann die Transformation des gesamten Zustandsvektors zusammengestellt werden.

$$z = \Phi(\vec{x}) = \begin{bmatrix} z_1 \\ \vdots \\ z_r \\ z_{r+1} \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_r \\ \eta_1 \\ \vdots \\ \eta_{n-r} \end{bmatrix} \quad (6.12)$$

Aus Gleichung 6.10 ist ersichtlich, dass die ersten r transformierten Zustände jeweils der zeitlichen Ableitung des vorherigen Zustandes entsprechen ($\xi_i = \dot{\xi}_{i-1}$). Zur Bestimmung des zeitlichen Verhaltens des transformierten Systems, also seiner Dynamik, werden die Transformationsvorschrift $\vec{x} = \Phi^{-1}(\vec{z})$ in Gleichung 6.10 eingesetzt und zur übersichtlicheren Darstellung zusätzlich folgende Substitutionen eingeführt:

$$a(\vec{\xi}, \vec{\eta}) = L_g \cdot L_f^{-1} \cdot h(\Phi^{-1}(\vec{z})) , \quad b(\vec{\xi}, \vec{\eta}) = L_f^r \cdot h(\Phi^{-1}(\vec{z})) \quad (6.13)$$

Für die zusätzlichen i Transformationen $r + 1 \leq i \leq n$ werden ebenso Ersatzgrößen gewählt:

$$q_i(\vec{\xi}, \vec{\eta}) = L_f \cdot \Phi_i(\Phi^{-1}(\vec{z})) , \quad p_i(\vec{\xi}, \vec{\eta}) = L_g \cdot \Phi_i(\Phi^{-1}(\vec{z})) . \quad (6.14)$$

Außerdem kann man die beiden Teilzustandsvektoren $\vec{\xi} = [\xi_1, \dots, \xi_r]^T$ und $\vec{\eta} = [\eta_1, \dots, \eta_{n-r}]^T$ einführen, so dass zur Beschreibung der Dynamik des transformierten Systems die sogenannte Normalform zusammengefasst werden kann [37].

$$\dot{\vec{\xi}} = \begin{bmatrix} \xi_2 \\ \vdots \\ \xi_r \\ b(\vec{\xi}, \vec{\eta}) + a(\vec{\xi}, \vec{\eta}) \cdot u \end{bmatrix} , \quad \dot{\vec{\eta}} = \vec{q}(\vec{\xi}, \vec{\eta}) + \vec{p}(\vec{\xi}, \vec{\eta}) \cdot u \quad (6.15)$$

Unter Berücksichtigung der gleichen nichtlinearen Zustandstransformation folgt für die Ausgangsgröße y des Systems:

$$y = h(\Phi^{-1}(\vec{z})) = z_1 = \xi_1 . \quad (6.16)$$

In [37] ist darüber hinaus nachgewiesen, dass man die Transformation $\Phi(\vec{x})$ so wählen kann, dass die Zeitableitungen der letzten $n - r$ Zustände nicht von der Eingangsgröße u beeinflusst werden. Um den Punkt \vec{x}_0 gilt demnach $L_g \cdot \Phi_i(\vec{x}) = 0$, so dass sich Gleichung 6.15 wie folgt vereinfacht:

$$\dot{\vec{\xi}} = \begin{bmatrix} \xi_2 \\ \vdots \\ \xi_r \\ b(\vec{\xi}, \vec{\eta}) + a(\vec{\xi}, \vec{\eta}) \cdot u \end{bmatrix}, \quad \dot{\vec{\eta}} = \vec{q}(\vec{\xi}, \vec{\eta}) . \quad (6.17)$$

Diese Gleichung wird auch eingangsnormalisierte Normalform genannt. Sowohl die Normalform als auch die eingangsnormalisierte Normalform lassen sich mit geringen Erweiterungen auch für ein Mehrgrößensystem mit m Ausgangsgrößen herleiten, was in [37] und [52] dargestellt ist. Dabei werden die m Ausgangsgrößen jeweils getrennt nach den Eingangsgrößen abgeleitet, bis ein expliziter Zusammenhang auftritt. Bei der Zustandstransformation entstehen dann m Teilzustandsvektoren der Form $\vec{\xi}^i = [\xi_1^i, \dots, \xi_{r_i}^i]^T$, die sich dann zu einem Gesamtvektor $\vec{\xi} = [\xi^1, \dots, \xi^m]^T$ zusammenführen lassen. Für die einzelnen Ausgangsgrößen des Systems folgt schließlich:

$$y_i = h_i(\Phi^{-1}(\vec{z})) = \xi_1^i . \quad (6.18)$$

Fasst man diese einzelnen Ausgänge nun in einem Vektor zusammen, ergibt sich der Zusammenhang gemäß Gleichung 6.19.

$$\begin{bmatrix} y_i^{(r_i)} \end{bmatrix}_{m \times 1} = \mathbf{b}(\vec{x}) + \mathbf{A}(\vec{x}) \cdot \vec{u} = \mathbf{b}(\vec{\xi}, \vec{\eta}) + \mathbf{A}(\vec{\xi}, \vec{\eta}) \cdot \vec{u} \quad (6.19)$$

6.1.2 Linearisierende Zustandsrückführung

Mit Hilfe der dargestellten Zustandstransformation ist es nun möglich, eine nichtlineare Zustandsrückführung zu implementieren, die dazu führt, dass das transformierte System ein lineares Eingangs-/Ausgangsverhalten aufweist. Durch Einführung der bereits erwähnten Ersatzregelgrößen $\vec{v} = [v_1, \dots, v_m]^T$, die jeweils nur eine Ausgangsgröße beeinflussen, wird außerdem die Entkopplung des Eingangs-/Ausgangsverhaltens des Systems erreicht. Unter Verwendung der Ersatzregelgrößen \vec{v} und in Anlehnung an die Nomenklatur aus [52] kann für ein Mehrgrößensystem folgende nichtlineare Zustandsrückführung gewählt werden:

$$\vec{u} = \vec{\alpha}(\vec{x}) + \vec{\beta}(\vec{x}) \cdot \vec{v} . \quad (6.20)$$

Damit muss für die vom Zustandsvektor abhängige Eingangsgröße ebenfalls die Koordinatentransformation aus Abschnitt 6.1.1 angewendet werden, so dass sich Gleichung 6.20 wie folgt umschreiben lässt:

$$\vec{u} = \vec{\alpha}(\vec{\xi}, \vec{\eta}) + \vec{\beta}(\vec{\xi}, \vec{\eta}) \cdot \vec{v}. \quad (6.21)$$

Wird dieser Zusammenhang nun in Gleichung 6.1 eingesetzt, kann man den Effekt der Zustandsrückführung veranschaulichen:

$$\dot{\vec{x}} = \vec{f}(\vec{x}) + \mathbf{G}(\vec{x}) \cdot \vec{\alpha}(\vec{x}) + \mathbf{G}(\vec{x}) \cdot \vec{\beta}(\vec{x}) \cdot \vec{v}. \quad (6.22)$$

Nach [37] ist es dabei nicht relevant, ob zuerst die Transformation der Zustände erfolgt und dann die Rückführung derselben oder umgekehrt. Setzt man die gewählte Zustandsrückführung nun in Gleichung 6.19 ein, ergibt sich für die Ausgangsgrößen des geschlossenen Regelkreises:

$$\left[y_i^{(r_i)} \right]_{m \times 1} = \mathbf{b}(\vec{x}) + \mathbf{A}(\vec{x}) \cdot \left[\vec{\alpha}(\vec{x}) + \vec{\beta}(\vec{x}) \cdot \vec{v} \right]. \quad (6.23)$$

Wählt man nun die Koeffizienten $\vec{\alpha}$ und $\vec{\beta}$ in geeigneter Weise, nämlich mit:

$$\vec{\alpha}(\vec{x}) = -\mathbf{A}(\vec{x})^{-1} \cdot \mathbf{b}(\vec{x}) \quad \text{sowie} \quad \vec{\beta}(\vec{x}) = \mathbf{A}^{-1}(\vec{x}), \quad (6.24)$$

reduziert sich die Dynamik des geschlossenen Regelkreises auf die einfache Form gemäß Gleichung 6.25.

$$\left[y_i^{(r_i)} \right]_{m \times 1} = \vec{v} \quad (6.25)$$

Die resultierende Dynamik des geschlossenen Regelkreises ist sowohl linear als auch regelbar, was durch die Kombination aus Zustandstransformation und -rückführung ermöglicht wird [37]. Tatsächlich bedeutet dies, dass die nichtlineare verkoppelte Dynamik des untransformierten Systems in eine Integratorkette zwischen den Ersatzregelgrößen v_i und den ihnen zugeordneten Ausgängen y_i überführt werden kann. Der dafür notwendige Eingangsvektor \vec{u} kann durch Einsetzen der Gleichung 6.24 in Gleichung 6.20 bestimmt werden.

$$\vec{u} = \mathbf{A}^{-1}(\vec{x}) \cdot \left[\vec{v} - \mathbf{b}(\vec{x}) \right] \quad (6.26)$$

Die besondere Wirkungsweise der dargestellten nichtlinearen Zustandsrückführung soll wegen ihrer Bedeutung zusätzlich anhand Abbildung 6.2, die an [52] angelehnt ist, für ein Mehrgrößensystem vereinfacht illustriert werden. Es ist erkennbar, dass durch die beschriebene ZustandsTransformation und -rückführung die Invertierung der eigentlichen Systemdynamik gelingt, die zur Verdeutlichung mit $\hat{\mathbf{F}}^{-1}(\vec{x}, \vec{v})$ gekennzeichnet ist. Damit lässt sich aus der Ersatzregelgröße \vec{v} die zur Erzeugung eines gewünschten Systemausganges \vec{y} nötige Stellgröße \vec{u} bestimmen. Demnach folgt im Idealfall für die resultierende Übertragungsfunktion zwischen Ersatzregelgröße \vec{v} und dem Vektor der abgeleiteten Ausgänge $\vec{y}_i^{(r_i)}$: $\mathbf{F}(s) = \hat{\mathbf{F}}^{-1}(\vec{x}, \vec{v}) \cdot \mathbf{F}(\vec{x}, \vec{u}) = 1$. In Abhängigkeit des relativen Grades des Gesamtsystems werden schließlich r Integratoren benötigt, um aus dem Vektor $\vec{y}_i^{(r_i)}$ die Ausgänge \vec{y} zu ermitteln.

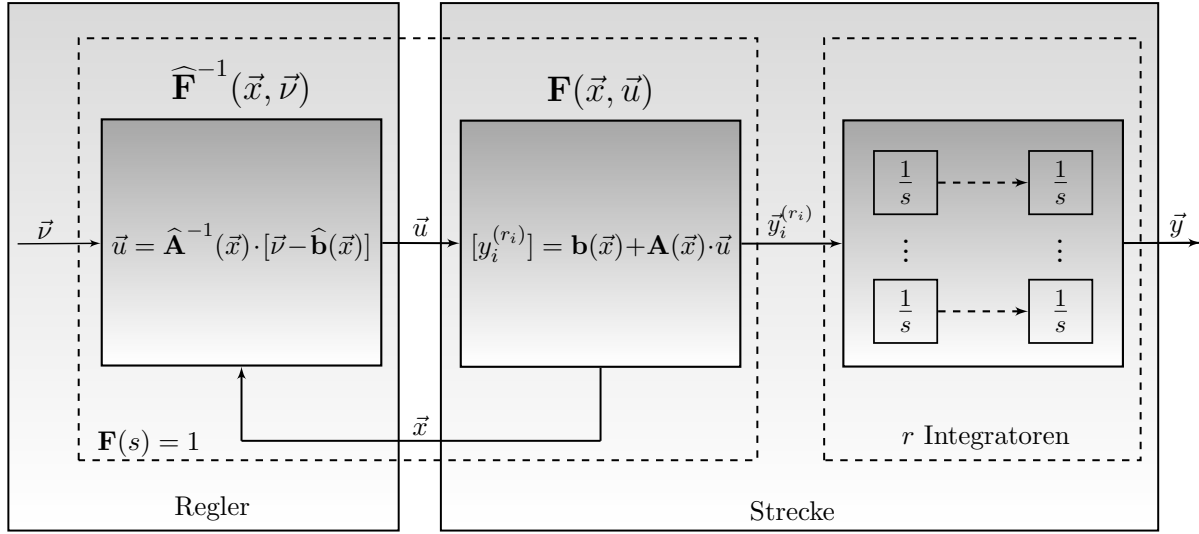


Abbildung 6.2: Darstellung der linearisierenden Zustandsrückführung.

6.1.3 Interne Dynamik des Regelkreises

In Abschnitt 6.1.1 sind im Rahmen der Zustandstransformation sowohl die Normalform als auch die eingangsnormalisierte Normalform eingeführt worden. Betrachtet man nun die Normalform aus Gleichung 6.15, erkennt man, dass der transformierte Zustandsvektor und damit auch seine Dynamik aus zwei Anteilen besteht, nämlich $\dot{\vec{\xi}}$ und $\dot{\vec{\eta}}$. Aus Gleichung 6.18 ist bekannt, dass ein Ausgang y_i von einem transformierten Zustand ξ_1^i abhängig ist. Das bedeutet im Umkehrschluss nichts anderes, als dass nur die Zustände des Teilzustandsvektors $\vec{\xi} = [\xi_1, \dots, \xi_r]^T$ über die jeweiligen Ausgänge regelungstechnisch beobachtbar sind [37, 52]. Da demzufolge das zeitliche Verhalten des Teilzustandsvektors $\vec{\eta} = [\eta_1, \dots, \eta_r]^T$ nicht über Systemausgänge zu quantifizieren ist, beschreibt dieser die interne Dynamik des Regelkreises.

Es liegt auf der Hand, dass die interne Dynamik kein destabilisierendes Verhalten aufweisen darf, damit eine adäquate Regelgüte erzielt wird. Ohne zu weit vorzugreifen, kann dies an der Regelung der Flugzeuglängsbewegung verdeutlicht werden; je nach gewählter Architektur der Lageregelung ist es möglich, dass der Anstellwinkel α Bestandteil des Teilzustandsvektors $\vec{\eta}$ und seine Änderung damit Teil der internen Dynamik wird. Es ist leicht ersichtlich, dass aus flugmechanischer Sicht eine unkontrollierte Dynamik des Anstellwinkels nicht wünschenswert ist. Um die interne Dynamik besser analysieren zu können, kann man den Ansatz der Null-dynamik heranziehen [37], wofür die eingangsnormalisierte Normalform aus Gleichung 6.17 betrachtet wird. Dabei sollen für den betrachteten Zeitraum t die Ausgänge des Systems $\vec{y}(t)$ im Ursprung gehalten werden, also gleich Null sein. Dies bedeutet folglich im Sinne der Zustandstransformation aus Gleichung 6.10 für ein Eingrößensystem, dass der Teilzustandsvektor $\vec{\xi} = [\xi_1, \dots, \xi_r]^T = 0$ ist $\forall t$. Setzt man diese Bedingungen in Gleichung 6.17 ein, lässt sich folgender Verlauf für die Eingangsgröße ableiten:

$$u(t) = -\frac{b(0, \vec{\eta}(t))}{a(0, \vec{\eta}(t))}, \quad (6.27)$$

wobei gilt, dass $y(0) = 0$ ist und $\vec{\eta}(0) = \vec{\eta}_0$ beliebig gewählt werden kann. Für die zeitliche Änderung der an der Nulldynamik beteiligten Zustände folgt außerdem:

$$\dot{\vec{\eta}} = \vec{q}(0, \vec{\eta}) , \quad (6.28)$$

so dass mit der Nulldynamik eine Beschreibung für das interne Systemverhalten gefunden ist, das von den Ausgangsgrößen entkoppelt ist. Dies hat den Vorteil, dass eine deutlich vereinfachte Untersuchung der Stabilität der internen Dynamik erfolgen kann, da nach [37] und [52] die Stabilität der Nulldynamik zur Folge hat, dass die Zustände der internen Dynamik beschränkt sind. Außerdem kann das Prinzip der Minimalphasigkeit linearer Systeme auf nichtlineare übertragen werden. Nach [72] ist ein Regelkreis ein Minimalphasenglied, wenn dies für den offenen Kreis gilt und wenn er zudem stabil ist. Die Aussage der Minimalphasigkeit ist demnach eine Aussage über die Stabilität des Systems, wobei für ein nichtlineares System festgestellt werden kann, dass es asymptotisch minimalphasig ist, wenn seine Nulldynamik asymptotisch stabil ist. Die Untersuchung der Stabilität der Nulldynamik hat demnach große Bedeutung für die Stabilität nichtlinearer Systeme.

6.1.4 Referenzmodelle

Aus den vorigen Darstellungen ist ersichtlich geworden, dass im Mehrgrößenfall der Vektor der Ersatzregelgrößen \vec{v} in das System geleitet wird und im Falle einer idealen Übertragungsfunktion $\mathbf{F}(s) = 1$ von Inversion und Strecke in den Vektor der abgeleiteten Ausgangsgrößen übergeht $\vec{y}_i^{(r_i)}$ (siehe Gleichung 6.25), die ihrerseits r mal zu den Systemausgängen aufintegriert werden. Soll nun eine Ausgangsgröße $y_i(t)$ einer bestimmten Wunschkurve folgen, so muss demzufolge für dieses Referenzsignal $y_{R,i}(t)$ in Abhängigkeit des relativen Grades auch die r -te Ableitung $y_{R,i}^{(r)}(t)$ ermittelt werden, bevor sie als Ersatzregelgröße \vec{v} in das System geleitet werden kann.

Zur Bestimmung der Referenzsignale und ihrer Ableitungen werden normalerweise lineare Referenzmodelle verwendet, wie sie bereits am Beispiel eines Systems zweiter Ordnung in Abschnitt 5.2.4 eingeführt worden sind. Für die dynamische Inversion gelten hinsichtlich ihrer Auslegung verständlicherweise die gleichen grundlegenden Annahmen wie für die modellfreie neuronale Regelung. Wie aus Abbildung 5.6 ersichtlich ist, können die nötigen Ableitungen gewonnen werden, indem man ihre Signale vor den jeweiligen Integratoren abgreift. In Abschnitt 6.4 wird gezeigt, dass es für die Flugregelungsanwendung sinnvoll sein kann, den Regelkreis kaskadisch mit drei Schleifen des relativen Grades von eins für Längs- und Seitenbewegung auszulegen. In diesem Falle vereinfacht sich das Referenzmodell zu einem PT_1 -Glied, in dem es nur die Zeitkonstante T als Entwurfsparameter auszulegen gilt.

Der in Abbildung 6.2 dargestellte Idealfall besagt folglich, dass eine Ausgabe $y_i(t)$ und ihre Ableitungen exakt der ihr zugeordneten Referenzgröße $y_{R,i}(t)$ und ihren Ableitungen folgen können. In der realen Anwendung wird dies auf Grund einer Reihe von Einflüssen allerdings

nicht der Fall sein. Dazu gehören unter anderem Modellunsicherheiten wegen nicht exakter Systemidentifizierung, auftretendes Messrauschen und Messfehler sowie Messverzögerungen hervorgerufen durch die Sensordynamik und nicht zuletzt externe Störungen und unter Umständen Schäden am System. In Einklang mit der in Abbildung 6.2 eingeführten Kennzeichnung soll das der Regelung zu Grunde liegende nichtlineare Modell für den Mehrgrößenfall wie folgt dargestellt werden:

$$\begin{aligned}\dot{\hat{\vec{x}}} &= \hat{\vec{f}}(\vec{x}) + \hat{\mathbf{G}}(\vec{x}) \cdot \vec{u} , \\ \hat{\vec{y}} &= \hat{\vec{h}}(\vec{x}) .\end{aligned}\quad (6.29)$$

In Anlehnung an Gleichung 6.19 ergibt sich für die Ableitungen der Ausgangsgrößen:

$$\left[\hat{\vec{y}}_i^{(r_i)} \right]_{m \times 1} = \hat{\mathbf{b}}(\vec{x}) + \hat{\mathbf{A}}(\vec{x}) \cdot \vec{u} . \quad (6.30)$$

Die linearisierende Zustandsrückführung basiert auf den Modellkenntnissen, so dass sich Gleichung 6.26 dementsprechend ändert.

$$\vec{u} = \hat{\mathbf{A}}^{-1}(\vec{x}) \cdot \left[\vec{v} - \hat{\mathbf{b}}(\vec{x}) \right] \quad (6.31)$$

Setzt man nun die Stellgröße aus Gleichung 6.31 in Gleichung 6.19 ein und nimmt vereinfachend an, dass keine zu berücksichtigende Stelldynamik vorliegt, so ergibt sich für das Eingangs-/Ausgangsverhalten des realen Systems:

$$\left[\vec{y}_i^{(r_i)} \right]_{m \times 1} = \mathbf{b}(\vec{x}) + \mathbf{A}(\vec{x}) \cdot \hat{\mathbf{A}}^{-1}(\vec{x}) \cdot \left[\vec{v} - \hat{\mathbf{b}}(\vec{x}) \right] . \quad (6.32)$$

Unter Berücksichtigung des Umstandes, dass der Referenzverlauf $\vec{y}_{R,i}^{(r_i)}$ in die Ersatzregelgröße \vec{v} übergeht, kann man den Inversionsfehler $\vec{\Delta}$ gemäß Gleichung 6.33 als Abweichung zwischen realem Ausgangsverhalten $\vec{y}_i^{(r_i)}$ und dazugehörigem Referenzverlauf ermitteln [52].

$$\vec{\Delta} = \left[\mathbf{A}(\vec{x}) \cdot \hat{\mathbf{A}}^{-1}(\vec{x}) - \mathbf{I} \right] \cdot \left[\vec{y}_{R,i}^{(r_i)} \right] + \left[\mathbf{b}(\vec{x}) - \mathbf{A}(\vec{x}) \cdot \hat{\mathbf{A}}^{-1}(\vec{x}) \cdot \hat{\mathbf{b}}(\vec{x}) \right] \quad (6.33)$$

Es ist leicht nachzuvollziehen, dass das Modell dem Referenzverlauf mit Sicherheit folgen kann, das heißt: $\vec{y}_{R,i}^{(r_i)} = \vec{v} = \hat{\vec{y}}_i^{(r_i)}$. Dadurch erhält man eine vereinfachte und leicht zu interpretierende Darstellung des Inversionsfehlers:

$$\vec{\Delta} = \left[\vec{y}_i^{(r_i)} \right]_{m \times 1} - \left[\vec{y}_{R,i}^{(r_i)} \right]_{m \times 1} = \left[\vec{y}_i^{(r_i)} \right]_{m \times 1} - \left[\hat{\vec{y}}_i^{(r_i)} \right]_{m \times 1} . \quad (6.34)$$

Der Inversionsfehler entspricht also den Abweichungen zwischen den Ableitungen der jeweiligen Ist- und Referenzverläufe. Da aber nun der Ausgangsvektor \vec{y} durch r -fache Integration der jeweiligen Ableitungen gebildet wird, kann man letztlich den Inversionsfehler als Abweichung zwischen genutztem Modell und realem System interpretieren. Es wird im weiteren Verlauf nun also das Ziel sein, den Einfluss des existierenden Inversionsfehler $\vec{\Delta}$ durch eine Kombination geeigneter Maßnahmen zu minimieren. Diese Maßnahmen sind die Fehlerkompensation durch einen linearen Regler, die dynamische Beschränkung der Ersatzregelgröße zur Berücksichtigung der Aktuatordynamik und die Approximation des verbleibenden Inversionsfehlers durch neuronale Netzwerke, wobei letztere wegen ihrer Lernfähigkeit auch auf Beschädigungen des Systems während des Betriebes zu reagieren vermögen.

6.1.5 Fehlerdynamik und linearer Regler

Bereits am Ende des Abschnittes 6.1.2 ist dargelegt worden, dass die Anwendung von Zustands-
transformation und -rückführung in einem sich linear verhaltenden Ersatzsystem resultiert, bei
dem die Ersatzregelgröße \vec{v} durch r -fache Integration in die Ausgangsgröße \vec{y} überführt wird.
Dieses Verhalten ist in Abbildung 6.3 stark vereinfacht illustriert.

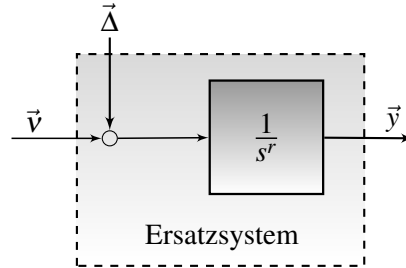


Abbildung 6.3: Darstellung des vereinfachten Ersatzsystems.

Soll nun ein linearer Regler implementiert werden, der den in Abschnitt 6.1.4 erläuterten Inver-
sionsfehler $\vec{\Delta}$ minimiert, so regelt dieser nicht mehr die eigentliche nichtlineare Strecke, sondern
das lineare Ersatzsystem. Nimmt man Bezug zur linearen Regelungstechnik, so entspricht der
Inversionsfehler einer auf das System einwirkenden Störung [99, 109]. In Anlehnung an die in
der Literatur gängige Nomenklatur [37, 40, 43, 52] lässt sich für den allgemeinen Fall eines
Mehrgrößensystems der Regelfehler definieren:

$$\vec{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_m \end{bmatrix} = \begin{bmatrix} y_{R,1} - y_1 \\ \vdots \\ y_{R,m} - y_m \end{bmatrix}. \quad (6.35)$$

Im weiteren Verlauf werden auch die Ableitungen des Regelfehlers benötigt, welche element-
weise nach der einfachen Vorschrift $e_i^{(j)} = y_1^{(j)} - y_{R,1}^{(j)}$ für alle m Fehler bestimmt werden kön-
nen. Zur übersichtlicheren Darstellung soll das Regelungsprinzip nun wie zuvor anhand eines
Eingrößensystems erläutert werden. Nutzt man die Zustandstransformation aus Abschnitt 6.1.1
auch für das Referenzsignal y_R und seine Ableitungen, so ergibt sich folgender Zusammenhang:

$$\vec{\xi}_R = \begin{bmatrix} \xi_{R,1} \\ \xi_{R,2} \\ \vdots \\ \xi_{R,r} \end{bmatrix} = \begin{bmatrix} y_R \\ \dot{y}_R \\ \vdots \\ y_R^{(r-1)} \end{bmatrix}. \quad (6.36)$$

Zur Bestimmung der Ersatzregelgröße v können nun neben der höchsten Ableitung der Refe-
renzgröße $y_R^{(r)}$ auch alle weiteren verfügbaren Abweichungen $e^{(i)} = y^{(i)} - y_R^{(i)} = \xi_{i-1} - \xi_{R,i-1}$
einbezogen werden, was in Gleichung 6.37 dargelegt ist.

$$v = y_R^{(r)} - c_0 \cdot (y - y_R) - c_1 \cdot (\dot{y} - \dot{y}_R) - \cdots - c_{r-1} \cdot (y^{(r-1)} - y_R^{(r-1)}) \quad (6.37)$$

Dieser Zusammenhang lässt sich mit Hilfe der Zustandstransformation verkürzt darstellen, wobei man erkennt, dass sich die Ersatzregelgröße v aus dem Anteil des Referenzverlaufes v_R und dem des linearen Reglers v_{LR} zusammensetzt:

$$v = y_R^{(r)} - \sum_{i=1}^r c_{i-1} \cdot (\xi_i - \xi_{R,i}) = y_R^{(r)} - \sum_{i=1}^{r-1} c_{i-1} \cdot e^{(i-1)} = v_R + v_{LR} . \quad (6.38)$$

Setzt man dies nun in Gleichung 6.21 für die linearisierende Zustandsrückführung ein und bedenkt, dass sich die Koeffizienten $\vec{\alpha}(\vec{\xi}, \vec{\eta})$ und $\vec{\beta}(\vec{\xi}, \vec{\eta})$ bei einem Eingrößensystem vereinfachen, so erhält man:

$$u = \frac{1}{a(\vec{\xi}, \vec{\eta})} \cdot \left[-b(\vec{\xi}, \vec{\eta}) + y_R^{(r)} - \sum_{i=1}^r c_{i-1} \cdot e^{(i-1)} \right] . \quad (6.39)$$

Benutzt man weiterhin aus Abschnitt 6.1.1 die Rücktransformationsvorschrift $\vec{x} = \Phi^{-1}(\vec{z})$ sowie die Substitutionen aus Gleichung 6.13, so kann man die Stellgröße u auch in Abhängigkeit der Zustandsgrößen \vec{x} darstellen:

$$u = \frac{1}{L_g \cdot L_f^{r-1} \cdot h(\vec{x})} \cdot \left[-L_f^r \cdot h(\vec{x}) + y_R^{(r)} - \sum_{i=1}^r c_{i-1} \cdot \left(L_f^{(i-1)} \cdot h(\vec{x}) - y_R^{(i-1)} \right) \right] . \quad (6.40)$$

Nun kann der Vektor $\vec{\chi}$ eingeführt werden, der sich aus dem Regelfehler und seinen $r-1$ Ableitungen zusammensetzt, was in Gleichung 6.41 erkennbar ist.

$$\vec{\chi}(t) = \left[e(t), \dot{e}(t), \dots, e^{(r-1)}(t) \right]^T \quad (6.41)$$

Unter Berücksichtigung, dass $\xi_i = y_R^{(r-1)} + e^{(i-1)}$ gilt, kann man Gleichung 6.39 in die eingangsnormalisierte Normalform aus Gleichung 6.17 einsetzen und erhält den Ausdruck

$$\begin{aligned} \dot{\vec{\chi}} &= \mathbf{A}_E \cdot \vec{\chi} , \\ \dot{\vec{\eta}} &= q(\vec{\xi}_R(t) + \vec{\chi}, \vec{\eta}) , \end{aligned} \quad (6.42)$$

wobei für die Matrix \mathbf{A}_E gilt:

$$\mathbf{A}_E = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -c_0 & -c_1 & -c_2 & \dots & -c_{r-1} \end{bmatrix} . \quad (6.43)$$

Damit kann schließlich der verändernde Einfluss der Reglerverstärkungen \vec{c} auf das bereits in Gleichung 6.7 beschriebene Eingangs-/Ausgangsverhalten des Systems charakterisiert werden:

$$y^{(r)} = y_R^{(r)} - c_0 \cdot e - c_1 \cdot \dot{e} - \dots - c_{r-1} \cdot e^{(r-1)} = y_R^{(r)} - \vec{c}^T \cdot \vec{\chi} . \quad (6.44)$$

Bezieht man weiterhin die Aussagen zu Gleichung 6.35 ein, so kann die Dynamik des Regelfehlers, im weiteren Verlauf Fehlerdynamik genannt, als Differentialgleichung r -ter Ordnung dargestellt werden:

$$e^{(r)} + c_{r-1} \cdot e^{(r-1)} + \dots + c_1 \cdot e + c_0 \cdot e = e^{(r)} + \vec{c}^T \cdot \vec{\chi} = 0. \quad (6.45)$$

Die Fehlerdynamik, also die zeitliche Änderung des Regelfehlers, kann folglich über die im Vektor \vec{c} zusammengefassten Reglerverstärkungen beeinflusst werden, wobei im Sinne der Stabilität der Fehlerdynamik gefordert ist, dass das aus Gleichung 6.45 resultierende Polynom nur negative Eigenwerte besitzt. Wie bereits erwähnt, soll der lineare Regler die äußere Störung in Form des Inversionsfehlers Δ minimieren. Setzt man dessen Definition aus Gleichung 6.34 nun für ein Eingrößensystem in Gleichung 6.44 ein, erhält man folgenden Ausdruck:

$$e^{(r)} + c_{r-1} \cdot e^{(r-1)} + \dots + c_1 \cdot e + c_0 \cdot e = \Delta. \quad (6.46)$$

Regelungstechnisch interpretiert erhält man mit Gleichung 6.46 ein lineares System r -ter Ordnung, bei dem der Inversionsfehler Δ als Eingangsgröße fungiert. Überträgt man diese Erkenntnis nun auf die Gleichung 6.42, so erhält man eine Zustandsraumdarstellung, die die zeitliche Änderung des Fehlervektors $\vec{\chi}$ beschreibt.

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} + \mathbf{b}_E \cdot \Delta \quad (6.47)$$

Hierbei hat der Vektor \mathbf{b}_E die Form $\mathbf{b}_E = [0, 0, \dots, 0, 1]^T$. Da diese Gleichung die Dynamik der zu minimierenden Fehlereinflüsse beschreibt, ist sie von großer Bedeutung und wird demzufolge in den folgenden Ausführungen wieder aufgegriffen und erweitert. Dies geschieht nicht zuletzt deswegen, weil die übersichtliche Darstellung als Zustandsraummodell die Interpretation des Systems erheblich vereinfacht. Hinsichtlich der Stabilität des geschlossenen Regelkreises sei angemerkt, dass zusätzlich zu der Forderung, dass die Elemente des Vektors \vec{c} positiv sein müssen, in [37] ein detaillierter Stabilitätsnachweis erbracht worden ist. Außerdem ist in [52] dargestellt, dass man für den Fehlervektor $\vec{\chi}$ bei vorgegebenem maximalen Inversionsfehler Δ eine obere Schranke bestimmen kann. Zusammenfassend ist nun in Abbildung 6.4 die Funktionsweise des erweiterten Regelkreises ähnlich wie in [52] und [54] für den Eingrößenfall dargestellt. Zusätzlich ist hier der Aktuator mit seiner Übertragungsfunktion $G_A(s)$ eingezeichnet, da die Berücksichtigung dessen dynamischen Verhaltens im weiteren Verlauf diskutiert wird. Diese Abbildung stellt die Grundkonfiguration der dynamischen Inversion dar, die zur Steigerung der Regelgüte erweitert wird. Zur besseren Nachvollziehbarkeit der vorzustellenden Erweiterungen wird Abbildung 6.4 in den folgenden Abschnitten sukzessive um die jeweiligen Elemente ergänzt werden.

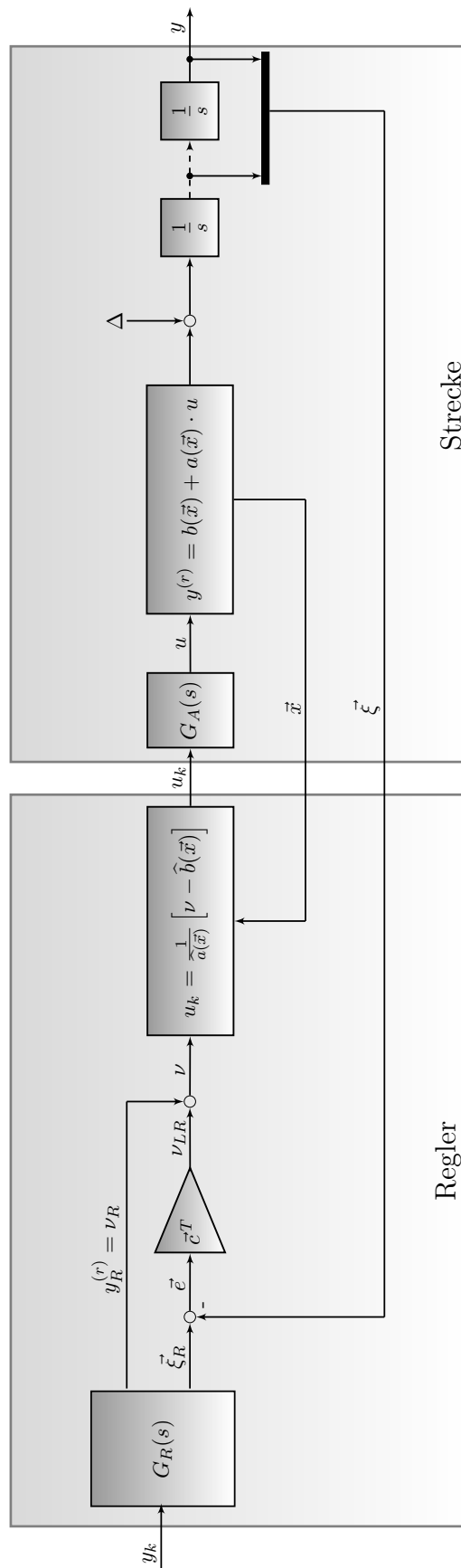


Abbildung 6.4: Darstellung des durch einen linearen Regler erweiterten Regelkreises.

6.2 Dynamische Beschränkung der Ersatzregelgrößen

In den bisherigen theoretischen Ausführungen ist zunächst davon ausgegangen worden, dass die Stellsignale \vec{u} exakt den kommandierten Werten \vec{u}_k folgen können. Da Aktuatoren aber ein eigenes dynamisches Verhalten besitzen, das häufig mit Systemen erster oder zweiter Ordnung beschrieben werden kann [28], sollte diese Dynamik beim Reglerentwurf berücksichtigt werden. Ein Ansatz wäre zweifellos, die Stelldynamik als Teil der Regelstrecke aufzufassen und somit bei der Inversion explizit zu berücksichtigen. Dies führt in Abhängigkeit der Ordnung der Stelldynamik aber zu einer entsprechenden Erhöhung des relativen Grades und damit zu einer nicht unbedingt wünschenswerten Steigerung der Komplexität des resultierenden Systems inklusive weiterer Probleme bei der Ermittlung der nötigen Ableitungen der Systemausgänge.

Eine Alternative hierzu bildet die bereits erwähnte dynamische Beschränkung der Ersatzregelgrößen, auch *pseudo control hedging* - PCH genannt. Das Prinzip besteht darin, den durch Vernachlässigung der Stelldynamik entstehenden Inversionsfehler abzuschätzen und die Dynamik des Referenzmodells anzupassen, es gewissermaßen mit Kenntnis über den aktuellen Flugzustand vorzusteuern. Eine gute Übersicht über das PCH ist in [42, 44, 52, 55] zu finden, wobei weiterführende und detailliertere Analysen in [43, 45, 48] dargestellt sind. Sind also Stellgrößen mess- oder schätzbar, kann man mit dieser Vorsteuerung erreichen, dass ein Referenzmodell nur Ausgaben erzeugt, denen das reale System trotz existierender Stelldynamik und -begrenzung wirklich folgen kann. Zur Verdeutlichung der Wirkungsweise ist das PCH in Abbildung 6.5 für ein Eingrößensystem dargestellt, wobei die in Abschnitt 6.1.4 eingeführte Notation zur Kennzeichnung von Modell und realem System beibehalten wird.

In Abbildung 6.5 ist noch einmal illustriert, dass ein Aktuator ein eigenes Übertragungsverhalten besitzt, so dass für die Berechnung der tatsächlichen Stellgröße $u = G_A(u_k)$ gilt. Ist die Stellgröße u bekannt, kann nun unter Zuhilfenahme des Modells $\hat{\mathbf{F}}(\vec{x}, u)$ die Reaktion des realen Systems anhand von Gleichung 6.48 geschätzt werden.

$$\hat{\mathbf{v}} = \hat{\mathbf{F}}(\vec{x}, u) = \hat{\mathbf{F}}(\vec{x}, G_A(u_k)) \quad (6.48)$$

Damit lässt sich die in Abbildung 6.5 erkennbare Differenz zwischen kommandierter und modellierter Systemantwort \mathbf{v}_h unter Zuhilfenahme des Zusammenhangs für die kommandierte Stellgröße $\hat{\mathbf{F}}^{-1}(\vec{x}, \mathbf{v}) = u_k$ ermitteln zu:

$$\mathbf{v}_h = \mathbf{v} - \hat{\mathbf{v}} = \hat{\mathbf{F}}(\vec{x}, u_k) - \hat{\mathbf{F}}(\vec{x}, u) = \mathbf{v} - \hat{\mathbf{F}}(\vec{x}, G_A(\hat{\mathbf{F}}^{-1}(\vec{x}, \mathbf{v}))) \quad (6.49)$$

Somit ist eine Größe beschrieben, die die Abweichung zwischen Vorgabe und Systemantwort auf Grund der begrenzten Stelldynamik quantifiziert. Die Dynamik des Referenzmodells wird nun insofern beeinflusst, als dass die r -te Ableitung $y_R^{(r)}$ um \mathbf{v}_h reduziert wird.

$$y_R^{(r)} = \mathbf{v}_R - \mathbf{v}_h \quad (6.50)$$

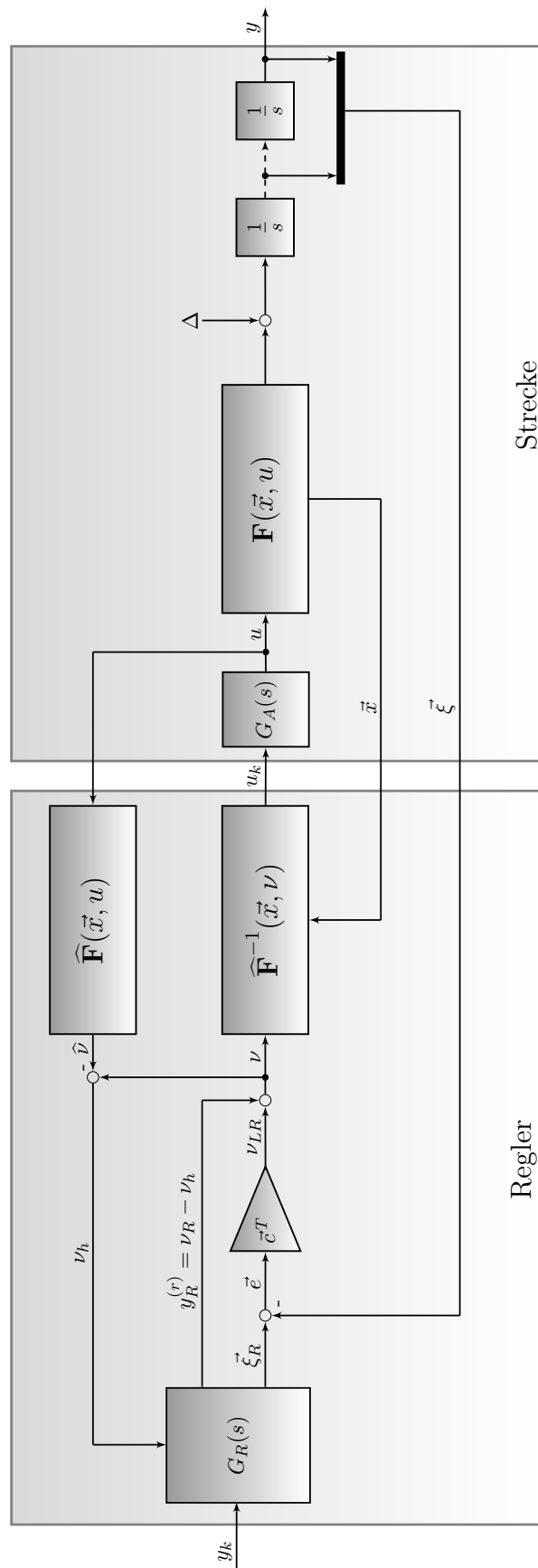


Abbildung 6.5: Darstellung des vorgesteuerten Referenzmodells im Regelkreis.

In Abbildung 6.6 ist nun das veränderte Übertragungsverhalten des Referenzmodells $G_R(s)$ an einem linearen System zweiter Ordnung verdeutlicht.

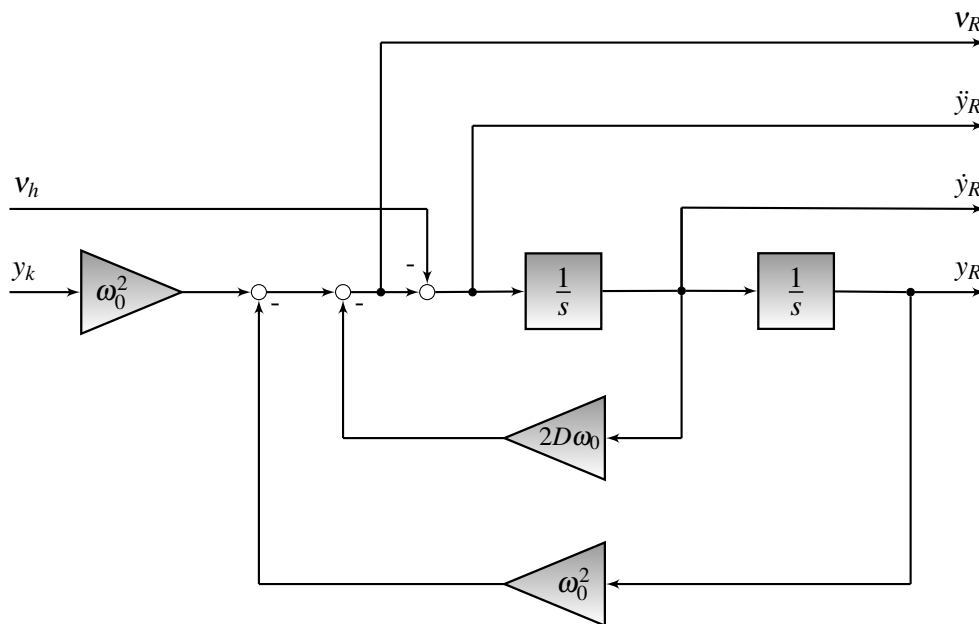


Abbildung 6.6: Durch Vorsteuerung beeinflusstes Referenzmodell zweiter Ordnung.

Führt man sich nun noch einmal das Ersatzsystem aus Abbildung 6.3 vor Augen und bedenkt, dass mit v_h eine zusätzliche Größe eingeführt wurde, die den Einfluss der Stelldynamik auf den Inversionsfehler Δ reduzieren soll, so kann für die Reaktion des Systems folgender Zusammenhang gefunden werden:

$$y^{(r)} = v + \Delta - v_h . \quad (6.51)$$

Bereits Gleichung 6.38 besagt, dass sich v aus dem Anteil von Regler und Referenzmodell zusammensetzt; bezieht man zusätzlich Gleichung 6.50 mit ein, so erhält man:

$$y^{(r)} = v_{LR} + \Delta + y_R^{(r)} . \quad (6.52)$$

Aus Gleichung 6.35 ist hingegen die Fehlerdefinition $e^{(r)} = y_R^{(r)} - y^{(r)}$ bekannt, so dass man nach Substitution von v_{LR} gemäß Gleichung 6.38 und Nutzung des Fehlerzustandsvektors $\vec{\chi}$ aus Gleichung 6.41 schließlich die Fehlerdynamik für die besagte Vorsteuerung des Referenzmodells erhält.

$$e^{(r)} + \vec{c}^T \cdot \vec{\chi} = \Delta \quad (6.53)$$

Damit zeigt sich, dass das *pseudo control hedging* keinen Einfluss auf die Fehlerdynamik besitzt und folglich auch nicht auf ihre Stabilität. Da der Linearregler die unbeeinflusste Fehlerdynamik regelt, zeigt sich daran auch, dass unerwünschte Effekte wie Stellraten- oder Ausschlagsbegrenzungen des Aktuators keine destabilisierenden Rückwirkungen auf diesen haben können; gleiches gilt auch für die im weiteren Verlauf zu erläuternde Verwendung neuronaler Netze zur

Approximation des Inversionsfehlers. Ein neuronales Netzwerk wird sich durch das PCH nicht an gegebenenfalls auftretende Ausschlagsbegrenzungen eines Aktuators anpassen, wodurch ein unbegrenztes Anwachsen der Verbindungsgewichte verhindert wird. Auch wenn die dynamische Begrenzung der Ersatzreglgröße augenscheinlich keinen Einfluss auf die Fehlerdynamik hat, so wird dadurch eine zusätzlich Rückführschleife eingeführt, durch die die Strecke auf das Referenzmodell wirkt. Damit muss dies in die Stabilitätsbetrachtung des Gesamtsystems einbezogen werden, wozu man zusätzlich den Fehler zwischen Referenzmodell und kommandierter Regelgröße betrachtet.

$$e_R = y_R - y_k \quad (6.54)$$

Dieser Ausdruck wird mit dem Fehler e aus Gleichung 6.35 zu einem Gesamtfehler e_k kombiniert:

$$e_k = e_R + e = y - y_k . \quad (6.55)$$

Unter Verwendung geeigneter Ljapunow-Funktionen wird in [45] und [48] gezeigt, dass beide Komponenten des Gesamtfehlers e_k beschränkt sind und somit das resultierende Gesamtsystem stabil ist.

6.3 Erweiterung durch neuronale Netzwerke

Es ist deutlich geworden, dass durch die Implementierung eines linearen Reglers und der dynamischen Begrenzung der Ersatzregelgröße der Einfluss des Inversionsfehlers auf den geschlossenen Regelkreis reduziert werden kann. Gleichwohl können je nach Grad der Systemkenntnis nicht unerhebliche Modellfehler verbleiben, für deren Kompensation sich die Lernfähigkeit neuronaler Netze anbietet. Durch deren Approximationseigenschaften können signifikante Parameterunsicherheiten ausgeglichen werden, so dass sich das Regelungssystem bis zu einem gewissen Grad auch an im Flug auftretende Systemschäden adaptieren kann. Dadurch erhält man ein gegenüber äußeren Störungen und Modellunsicherheiten robust agierendes, adaptives Regelungssystem.

Wie in der Einleitung zu diesem Kapitel dargelegt, ist dieser Ansatz schon mehrfach erfolgreich für die Flugregelungsanwendung zum Einsatz gekommen. Dabei werden vorwärtsgerichtete Netzwerke verwendet, die mit dem Rückpropagationsalgorithmus aus Kapitel 2 oder anwendungsspezifischen Erweiterungen dieses Algorithmus' trainiert werden [39, 45, 47]. Im weiteren Verlauf soll in Anlehnung an [52] zunächst der etablierte Ansatz aus [39] dargestellt werden, der im Sinne der verbesserten Stabilitätsbetrachtung des neuronalen Netzwerkes die in [119] vorgestellte e-Modifikation verwendet. Anschließend soll die Fehlerapproximation durch ein KNN um das in Kapitel 3 vorgestellte Lernverfahren basierend auf dem Konzept der strukturvariablen Regelung erweitert werden. Dessen Anwendbarkeit für die vorgestellte Regelungsstrategie ist in [106] untersucht worden und in [81] für ein unbemanntes Flugsystem dargestellt. Dabei wird sich zeigen, dass die stabilitätsbedingte Definition der Lernregeln die

Fehlerzustände des Netzwerkes und damit die Zustände der Fehlerdynamik des zu regelnden Systems in einen Gleitzustand zu zwingen vermag. Damit wird die Stabilitätsanalyse des Gesamtsystems deutlich erleichtert, da die in Kapitel 3 erläuterte Stabilitätsforderung des Lernalgorithmus' die Stabilität der durch das neuronale Netzwerk beeinflussten Fehlerdynamik bedingt.

Ziel der Nutzung neuronaler Netze ist es also, den verbleibenden Inversionsfehler auszugleichen, was letztlich der in Kapitel 2 erläuterten Abbildung eines dynamischen Prozesses entspricht. Bedient man sich nun der universellen Approximationseigenschaften eines KNN aus Abschnitt 2.4, so kann man Gleichung 2.39 auf den Inversionsfehler beziehen. Im Sinne der übersichtlicheren Darstellung wird für die Herleitung erneut ein Eingrößensystem betrachtet.

$$\Delta = \mathbf{w}_*^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}_*^{(1)} \vec{x}) + \varepsilon_A(\vec{x}) \quad (6.56)$$

Damit wird deutlich, dass die Ersatzregelgröße v erweitert werden muss, was in Gleichung 6.57 dargestellt ist.

$$v = v_R + v_{LR} - v_h - v_{ad} - v_r \quad (6.57)$$

Dabei ist der adaptive Term v_{ad} die Ausgangsgröße des KNN, durch die der Inversionsfehler Δ reduziert werden soll:

$$v_{ad} = \mathbf{w}^{(2)} \cdot \vec{f}^{(2)}(\mathbf{w}^{(1)} \vec{x}) . \quad (6.58)$$

Der Ausdruck v_r wird auch robustifizierender Term genannt. Er stellt im Grunde eine proportionale Fehlerrückführung dar und ist für die spätere Stabilitätsbetrachtung von Bedeutung:

$$v_r = [k_{r0} + k_{r1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*)] \cdot \vec{\zeta} . \quad (6.59)$$

Hierbei ist $\vec{\zeta}$ die sogenannte gefilterte Fehlergröße, die im weiteren Verlauf noch näher diskutiert wird, k_{r0} und k_{r1} sind positive Entwurfparameter, die für die Stabilität von Relevanz sind, $\|\mathbf{w}\|_F$ bezeichnet die Frobeniusnorm der Gesamtgewichtsmatrix \mathbf{w} aus Gleichung 2.6 und $\bar{\mathbf{w}}_*$ bezeichnet die obere Schranke der optimalen Verbindungsgewichte. Die Frobeniusnorm wird gemäß Gleichung 6.60 berechnet:

$$\|\mathbf{w}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n w_{ij}^2} . \quad (6.60)$$

Für die obere Schranke der optimalen Verbindungsgewichte, die ebenfalls für die Stabilitätsbetrachtung von Bedeutung ist, kann folgender Zusammenhang festgehalten werden:

$$\|\mathbf{w}_*\|_F \leq \bar{\mathbf{w}}_* . \quad (6.61)$$

Da es sich bei der Approximation des Inversionsfehlers um die Abbildung eines dynamischen Prozesses handelt, ist die Zusammensetzung des Eingangsvektors \vec{x} des neuronalen Netzes von

Bedeutung. Er sollte sich zumindest aus der Referenztrajektorie $\vec{\xi}_R(t)$ sowie messtechnisch verfügbaren, für die Modellierung des Inversionsfehlers relevanten Zustandsgrößen \vec{x} zusammensetzen und kann ebenso den Fehlervektor $\vec{\chi}$ enthalten:

$$\vec{x} = \begin{bmatrix} \vec{\xi}_R \\ \vec{\chi} \\ \vec{x} \end{bmatrix}. \quad (6.62)$$

Da es sich um einen zeitabhängigen Prozess handelt, ist darüber hinaus in Erwägung zu ziehen, auch zeitverzögerte Größen gemäß Abschnitt 2.5 in den Eingangsvektor \vec{x} aufzunehmen, um die Abbildungsqualität zu erhöhen. Mit der Zusammensetzung des Eingangsvektors sind nun alle grundlegenden Informationen bekannt, die die strukturelle Erweiterung des Regelkreises beschreiben. Zur Verdeutlichung ist diese nun in Abbildung 6.7 illustriert, so dass sich ein Gesamtsystem aus dynamischer Inversion, linearem Regler, Vorsteuerung der Referenzmodelle und neuronalen Netzwerken zur Minimierung von Modellfehlern ergibt. In Anlehnung an [45] und [52] können außerdem folgende erläuternde Ergänzungen gemacht werden. Aus der Forderung, dass der Inversionsfehler Δ durch das KNN reduziert werden soll, kann man die mathematische Bedingung $|\partial\Delta/\partial v_{ad}| < 1$ ableiten. Führt man an diesem Term einige Erweiterungen durch, so lassen sich zwei Forderungen an den Regelkreis aufstellen, nämlich dass das Vorzeichen der Wirkung einer Stellgröße im Modell mit dem realen System übereinstimmt und dass die im Modell angenommene Wirkung einer Stellgröße nicht geringer als 50 % der realen Stellgröße sein darf. Es ist ersichtlich, dass beide Forderungen an das Modell vergleichsweise einfach garantiert werden können.

6.3.1 Fehlerdynamik der erweiterten Reglerarchitektur

Wie schon zuvor stellt sich an dieser Stelle die Frage, inwieweit die Fehlerdynamik des geschlossenen Regelkreises durch die vorgenommene Erweiterung beeinflusst wird. Um dies näher zu analysieren, wird Gleichung 6.47 um die relevanten Terme der neuronalen Reglererweiterung ergänzt, so dass man folgenden Zusammenhang erhält:

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} + \mathbf{b}_E \cdot (\Delta - v_{ad} - v_r). \quad (6.63)$$

Führt man sich nun noch einmal das grundlegende Ziel der Approximation des Inversionsfehlers aus Gleichung 6.56 vor Augen, ist zu erkennen, dass es bei nicht optimalen Gewichten zu einem Restfehler in der Approximation kommen muss, der sich wie folgt darstellen lässt [39]:

$$\Delta - v_{ad} = \mathbf{w}_*^{(2)} \cdot \vec{f}(\mathbf{w}_*^{(1)} \vec{x}) - \mathbf{w}^{(2)} \cdot \vec{f}(\mathbf{w}^{(1)} \vec{x}) + \varepsilon_A(\vec{x}). \quad (6.64)$$

Dazu sei angemerkt, dass der Index der Transferfunktionen der verborgenen Schicht $\vec{f}^{(2)}$ zur übersichtlicheren Darstellung ausgelassen wird. Gleiches gilt für die Diagonalmatrix der Ableitungen der Transferfunktion, die im folgenden mit \mathbf{f}' gekennzeichnet ist.

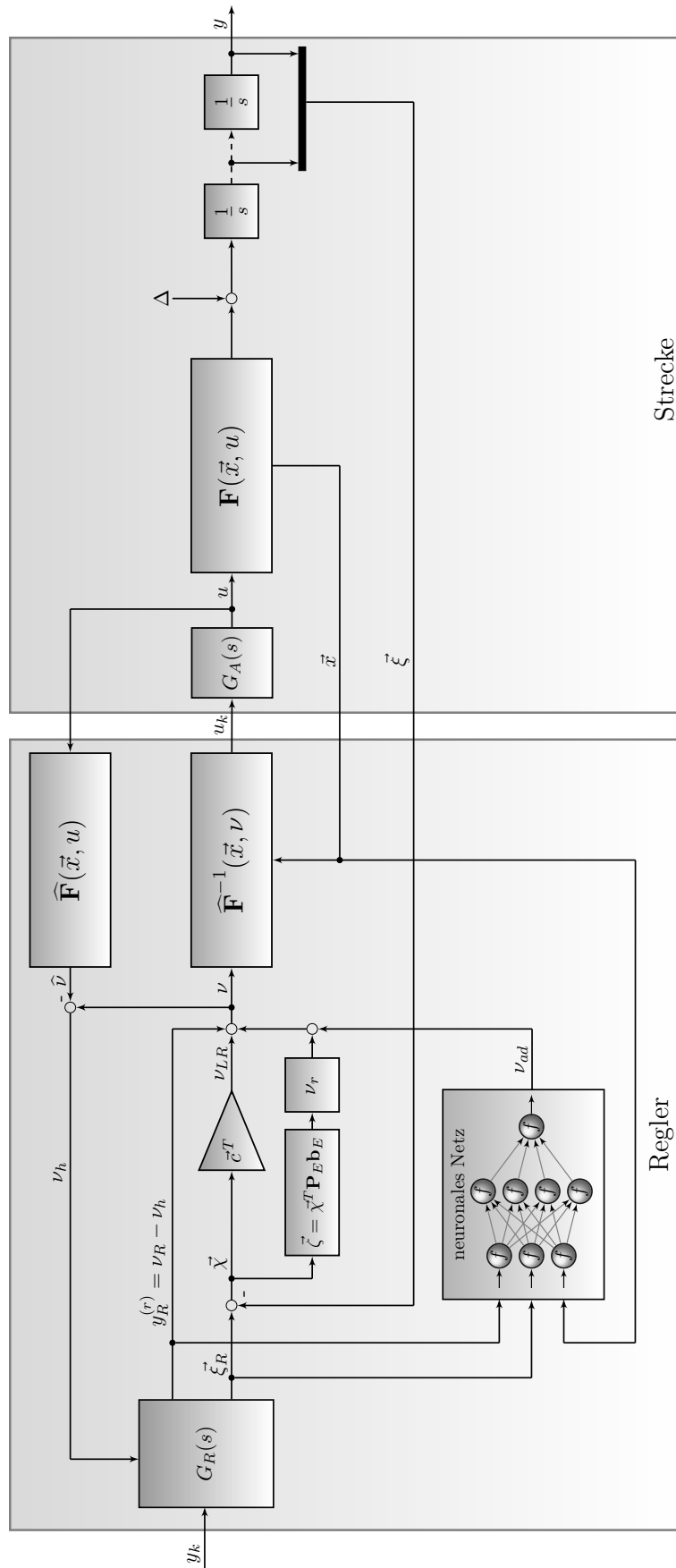


Abbildung 6.7: Darstellung des durch ein neuronales Netzwerk erweiterten Regelkreises.

Darüber hinaus bietet es sich an, Gewichtsfehler als Abweichungen zwischen den optimalen Gewichtsmatrizen $\mathbf{w}_*^{(1)}$, $\mathbf{w}_*^{(2)}$ und \mathbf{w}_* sowie deren aktuellen Werten $\mathbf{w}^{(1)}$, $\mathbf{w}^{(2)}$ und \mathbf{w} zu definieren. Hierbei sei angemerkt, dass diese Definition auch wie in [52] vorgenommen werden kann, was in einigen Vorzeichenänderungen der weiteren Ausführungen resultiert, die Ergebnisse insgesamt aber nicht ändert.

$$\tilde{\mathbf{w}}^{(1)} = \mathbf{w}^{(1)} - \mathbf{w}_*^{(1)}, \quad \tilde{\mathbf{w}}^{(2)} = \mathbf{w}^{(2)} - \mathbf{w}_*^{(2)}, \quad \tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}_* \quad (6.65)$$

Im Sinne der Stabilität der Fehlerdynamik ist es wichtig, den Restfehler der Approximation zu quantifizieren. Nach [39] ist dabei besonders die Beschreibung des Ausgabefehlers der verdeckten Schicht von Interesse, da nur in ihr nichtlineare Transferfunktionen verwendet werden. Dazu wird eine Taylorreihenentwicklung der verdeckten Schicht für einen gegebenen Eingangsvektor \vec{x} durchgeführt und nach dem ersten Glied abgebrochen:

$$\vec{f}(\mathbf{w}_*^{(1)}\vec{x}) = \vec{f}(\mathbf{w}^{(1)}\vec{x}) + \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot [\mathbf{w}_*^{(1)}\vec{x} - \mathbf{w}^{(1)}\vec{x}] + O[\mathbf{w}_*^{(1)}\vec{x} - \mathbf{w}^{(1)}\vec{x}]^2. \quad (6.66)$$

Setzt man den zuvor definierten Zusammenhang $\tilde{\mathbf{w}}^{(1)} = \mathbf{w}^{(1)} - \mathbf{w}_*^{(1)}$ für den Gewichtsfehler ein, kann man den Ausdruck wie folgt vereinfachen:

$$\vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) = \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot [\tilde{\mathbf{w}}^{(1)}\vec{x}] + O[\tilde{\mathbf{w}}^{(1)}\vec{x}]^2. \quad (6.67)$$

Durch die Entwicklung einer Taylorreihe ist im Grunde eine Linearisierung des Netzwerkes erfolgt, dessen Fehler durch Umstellen von Gleichung 6.67 beschrieben werden kann mit:

$$O[\tilde{\mathbf{w}}^{(1)}\vec{x}]^2 = \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot [\tilde{\mathbf{w}}^{(1)}\vec{x}]. \quad (6.68)$$

Diese Informationen können nun genutzt werden, um den verbleibenden Approximationsfehler und damit die Fehlerdynamik des durch das neuronale Netzwerk beeinflussten Systems näher zu beschreiben. Dazu wird Gleichung 6.64 herangezogen und durch geeignetes Erweitern und Einsetzen der zuvor definierten Gewichtsfehler umgeformt - zunächst ergibt sich:

$$\Delta - v_{ad} = \mathbf{w}_*^{(2)} \vec{f}(\mathbf{w}_*^{(1)}\vec{x}) - \mathbf{w}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) + \varepsilon_A(\vec{x}) + \mathbf{w}_*^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{w}_*^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}). \quad (6.69)$$

Mit der Definition der Gewichtsfehler vereinfacht sich der Ausdruck zunächst und kann erneut erweitert werden:

$$\Delta - v_{ad} = \underbrace{\mathbf{w}_*^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{w}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x})}_{-\tilde{\mathbf{w}}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x})} + \underbrace{\mathbf{w}_*^{(2)} \vec{f}(\mathbf{w}_*^{(1)}\vec{x}) - \mathbf{w}_*^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x})}_{-\mathbf{w}_*^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x})} + \varepsilon_A(\vec{x}), \quad (6.70)$$

$$\Delta - v_{ad} = -\tilde{\mathbf{w}}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{w}_*^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) + \mathbf{w}^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) - \mathbf{w}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) + \varepsilon_A(\vec{x}). \quad (6.71)$$

Erneutes Einsetzen des entsprechenden Gewichtsfehlers liefert:

$$\Delta - v_{ad} = -\tilde{\mathbf{w}}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{w}^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) - \underbrace{\mathbf{w}_*^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) + \mathbf{w}^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x})}_{\tilde{\mathbf{w}}^{(2)} \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x})} + \varepsilon_A(\vec{x}), \quad (6.72)$$

$$\Delta - v_{ad} = -\tilde{\mathbf{w}}^{(2)} \cdot \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{w}^{(2)} \cdot \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) + \tilde{\mathbf{w}}^{(2)} \cdot \vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x}) + \varepsilon_A(\vec{x}). \quad (6.73)$$

Hier ist zu erkennen, dass das Ergebnis aus Gleichung 6.67 eingesetzt werden kann, das die Taylorreihenentwicklung berücksichtigt.

$$\Delta - v_{ad} = -\tilde{\mathbf{w}}^{(2)} \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{w}^{(2)} \underbrace{\left[\mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \left[\tilde{\mathbf{w}}^{(1)}\vec{x} \right] + O \left[\tilde{\mathbf{w}}^{(1)}\vec{x} \right]^2 \right]}_{\vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x})} + \tilde{\mathbf{w}}^{(2)} \underbrace{\left[\mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \left[\tilde{\mathbf{w}}^{(1)}\vec{x} \right] + O \left[\tilde{\mathbf{w}}^{(1)}\vec{x} \right]^2 \right]}_{\vec{f}(\tilde{\mathbf{w}}^{(1)}\vec{x})} + \varepsilon_A(\vec{x}) \quad (6.74)$$

Nach entsprechendem Umformen und Ausmultiplizieren kann man unter Einführung eines sogenannten Störterms \mathbf{z} den Restfehler gemäß Gleichung 6.75 zusammenfassen.

$$\Delta - v_{ad} = \mathbf{z} - \tilde{\mathbf{w}}^{(2)} \cdot \left[\vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot \mathbf{w}^{(1)}\vec{x} \right] - \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot \tilde{\mathbf{w}}^{(1)}\vec{x} \quad (6.75)$$

Der Störterm \mathbf{z} ist dabei folgendermaßen definiert:

$$\mathbf{z} = \varepsilon_A(\vec{x}) - \tilde{\mathbf{w}}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \mathbf{w}_*^{(1)}\vec{x} + \mathbf{w}_*^{(2)} \cdot \vec{O}[\tilde{\mathbf{w}}^{(1)}\vec{x}]^2. \quad (6.76)$$

Damit sind nun alle Größen bekannt, um die von den neuronalen Netzwerken beeinflusste Fehlerdynamik aus Gleichung 6.63 genau zu beschreiben.

$$\dot{\vec{\chi}} = \mathbf{A}_E \cdot \vec{\chi} - \mathbf{b}_E \cdot \left[\mathbf{z} - \tilde{\mathbf{w}}^{(2)} \cdot \left[\vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \mathbf{w}^{(1)}\vec{x} \right] - \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \tilde{\mathbf{w}}^{(1)}\vec{x} - v_r \right] \quad (6.77)$$

Es liegt auf der Hand, dass es die Stabilität dieser Fehlerdynamik nachzuweisen gilt. Um dies zu tun, können in Anlehnung an [39] Lernregeln bestimmt werden, die den aus Abschnitt 2.3.1 bekannten Rückpropagationsalgorithmus erweitern. Diese Lernregeln können für die Gewichtsmatrizen $\mathbf{w}^{(1)}$ und $\mathbf{w}^{(2)}$ wie folgt definiert werden:

$$\Delta \mathbf{w}^{(1)} = \dot{\mathbf{w}}^{(1)} = \Gamma^{(1)} \cdot \left[\vec{x} \cdot \vec{\zeta} \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)T} \right], \quad (6.78)$$

$$\Delta \mathbf{w}^{(2)} = \dot{\mathbf{w}}^{(2)} = \Gamma^{(2)} \cdot \left[\vec{f}(\mathbf{w}^{(1)}\vec{x}) \cdot \vec{\zeta} - \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot \mathbf{w}^{(1)}\vec{x} \cdot \vec{\zeta} - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)T} \right]. \quad (6.79)$$

Hierbei sind $\Gamma^{(1)}$ und $\Gamma^{(2)}$ Matrizen, die die Lernraten der jeweiligen Schicht enthalten und gemäß Abschnitt 2.3.1 in der Regel die einheitliche Lernrate μ nutzen. Betrachtet man die Gleichungen 6.78 und 6.79, so kann man feststellen, dass der jeweils erste Term dem klassischen Rückpropagationsalgorithmus aus Gleichung 2.25 entspricht. Dabei ist die gefilterte Fehlergröße $\vec{\zeta}$ mit dem Ausgabefehler des Netzwerkes $\vec{\varepsilon}$ aus Gleichung 2.11 gleichzusetzen, was für die Anwendung des in Kapitel 3 vorgestellten Lernverfahrens noch von Bedeutung sein wird. Der zweite Term aus Gleichung 6.79 wird eingeführt, um den bei der Taylorreihenentwicklung

gemäß Gleichung 6.68 entstehenden Fehler im Sinne der Stabilität der Fehlerdynamik zu kompensieren. Der jeweils letzte Term bezeichnet die bereits erwähnte e-Modifikation, wobei über den Faktor λ die Verbindungsgewichte nach oben begrenzt werden können [119]. In [39] ist dies anschaulich als *Term des Vergessens* bezeichnet, da durch die e-Modifikation die Gewichte über die Zeit auch bei geringer externer Anregung durch Fehler gegen Null gebracht werden. Der Ausdruck $\|\vec{\zeta}\|_2$ wird als euklidische Norm bezeichnet, mit der die Länge eines Vektors beschrieben wird. Sie ist definiert gemäß Gleichung 6.80 und wird, ähnlich wie die Frobeniusnorm aus Gleichung 6.60 für Matrizen, bei Stabilitätsbetrachtungen herangezogen, um eine maximale obere Schranke für einen unbekannten Vektor nachzuweisen [39, 52, 77, 119].

$$\|\vec{\zeta}\|_2 = \sqrt{\vec{\zeta} \cdot \vec{\zeta}} = \sqrt{\zeta_1^2 + \zeta_2^2 + \dots + \zeta_n^2} \quad (6.80)$$

Die gefilterte Fehlergröße $\vec{\zeta}$, die in jedem Term der Lerngesetze enthalten ist, wird gemäß des Zusammenhanges $\vec{\zeta} = \vec{\chi}^T \cdot \mathbf{P}_E \cdot \mathbf{b}_E$ gebildet. Dabei ist \mathbf{b}_E der Eingangsvektor der linearen Fehlerdynamik aus Gleichung 6.63 und \mathbf{P}_E kann in Anlehnung an [77, 78] aus der folgenden Ljapunow-Gleichung gewonnen werden:

$$\mathbf{A}_E^T \cdot \mathbf{P}_E + \mathbf{P}_E \cdot \mathbf{A}_E = -\mathbf{Q}_E. \quad (6.81)$$

Hierbei ist \mathbf{A}_E die Dynamikmatrix der Fehlerdynamik und \mathbf{P}_E hat demnach die gleiche Dimension; die Matrix \mathbf{Q}_E kann frei vorgegeben werden, zum Beispiel als Einheitsmatrix. Zur Bestimmung von \mathbf{P}_E und einer oberen Schranke für den Zustandsvektor \vec{x} des betrachteten Systems kann folgende Ljapunow-Funktion gewählt werden:

$$V(\vec{x}) = \vec{x}^T \cdot \mathbf{P}_E \cdot \vec{x}. \quad (6.82)$$

Im Sinne der Stabilität muss die zeitliche Ableitung der Funktion $\dot{V}(\vec{x})$ negativ definit sein. Führt man die Ableitung durch und greift auf die Zustandsraumdarstellung des linearen Systems zurück, erhält man folgenden Zusammenhang:

$$\dot{V}(\vec{x}) = \dot{\vec{x}}^T \cdot \mathbf{P}_E \cdot \vec{x} + \vec{x}^T \cdot \mathbf{P}_E \cdot \dot{\vec{x}}, \quad (6.83)$$

$$\dot{V}(\vec{x}) = [\mathbf{A}_E \cdot \vec{x} + \mathbf{b}_E \cdot \vec{u}]^T \cdot \mathbf{P}_E \cdot \vec{x} + \vec{x}^T \cdot \mathbf{P}_E \cdot [\mathbf{A}_E \cdot \vec{x} + \mathbf{b}_E \cdot \vec{u}]. \quad (6.84)$$

Multipliziert man dies aus und setzt Gleichung 6.81 ein, kann folgende Zwischenlösung gefunden werden:

$$\dot{V}(\vec{x}) = -\vec{x}^T \cdot \mathbf{Q}_E \cdot \vec{x} + 2 \cdot \vec{x}^T \cdot \mathbf{P}_E \cdot \mathbf{b}_E \cdot \vec{u}. \quad (6.85)$$

Da $\dot{V}(\vec{x}) < 0$ gelten muss, kann unter Bestimmung einer oberen Schranke für den Zustandsvektor \vec{x} und den Systemeingang \vec{u} eine Lösung für die Matrix \mathbf{P}_E gefunden werden, was in [52] detailliert erläutert ist.

6.3.2 Stabilitätsbetrachtung des erweiterten Systems

Im weiteren Verlauf soll die Stabilität des durch neuronale Netzwerke erweiterten Regelkreises untersucht werden. Dazu soll, wie bei nichtlinearen Systemen üblich, die direkte Methode von Ljapunow herangezogen werden [39, 43, 45, 52, 77]. In Anlehnung an [39] und [52] wird eine positiv definite Ljapunow-Funktion gewählt, die den Einfluss der Fehlerdynamik des geschlossenen Regelkreises sowie die auftretenden Gewichtsfehler des KNN einbezieht, was in Gleichung 6.86 erkennbar ist.

$$V(\vec{\chi}) = \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \Gamma^{(1)-1} \tilde{\mathbf{w}}^{(1)T} \right) + \frac{1}{2} \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \Gamma^{(2)-1} \tilde{\mathbf{w}}^{(2)T} \right) \quad (6.86)$$

Da erneut gelten muss, dass $\dot{V}(\vec{\chi}) < 0$, wird die Zeitableitung von Gleichung 6.86 gebildet.

$$\begin{aligned} \dot{V}(\vec{\chi}) = & \frac{1}{2} \dot{\vec{\chi}}^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \dot{\vec{\chi}} + \frac{1}{2} \text{Spur} \left(\dot{\tilde{\mathbf{w}}}^{(1)} \Gamma^{(1)-1} \tilde{\mathbf{w}}^{(1)T} \right) + \frac{1}{2} \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \Gamma^{(1)-1} \dot{\tilde{\mathbf{w}}}^{(1)T} \right) \\ & + \frac{1}{2} \text{Spur} \left(\dot{\tilde{\mathbf{w}}}^{(2)} \Gamma^{(2)-1} \tilde{\mathbf{w}}^{(2)T} \right) + \frac{1}{2} \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \Gamma^{(2)-1} \dot{\tilde{\mathbf{w}}}^{(2)T} \right) \end{aligned} \quad (6.87)$$

Greift man nun auf die Gewichtsfehler aus Gleichung 6.65 zurück und leitet diese nach der Zeit ab, so erkennt man, dass sich der resultierende Ausdruck mit den Lernregeln aus den Gleichungen 6.78 und 6.79 darstellen lässt:

$$\dot{\tilde{\mathbf{w}}}^{(1)} = \frac{\partial(\tilde{\mathbf{w}}^{(1)})}{\partial t} = \frac{\partial(\mathbf{w}^{(1)} - \mathbf{w}_*^{(1)})}{\partial t} = \dot{\mathbf{w}}^{(1)}, \quad (6.88)$$

$$\dot{\tilde{\mathbf{w}}}^{(2)} = \frac{\partial(\tilde{\mathbf{w}}^{(2)})}{\partial t} = \frac{\partial(\mathbf{w}^{(2)} - \mathbf{w}_*^{(2)})}{\partial t} = \dot{\mathbf{w}}^{(2)}. \quad (6.89)$$

Setzt man diese Erkenntnisse in Gleichung 6.87 ein, kann folgende vereinfachte Darstellung gefunden werden:

$$\dot{V}(\vec{\chi}) = \frac{1}{2} \dot{\vec{\chi}}^T \mathbf{P}_E \vec{\chi} + \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \dot{\vec{\chi}} + \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \Gamma^{(1)-1} \dot{\mathbf{w}}^{(1)T} \right) + \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \Gamma^{(2)-1} \dot{\mathbf{w}}^{(2)T} \right). \quad (6.90)$$

In diese Gleichung werden nun die Fehlerdynamik aus Gleichung 6.63 und die Lernregeln eingesetzt.

$$\begin{aligned} \dot{V}(\vec{\chi}) = & \frac{1}{2} \left[\mathbf{A}_E \vec{\chi} + \mathbf{b}_E \cdot [\Delta - \mathbf{v}_{ad} + \mathbf{v}_r] \right]^T \cdot \mathbf{P}_E \vec{\chi} + \frac{1}{2} \vec{\chi}^T \mathbf{P}_E \cdot \left[\mathbf{A}_E \vec{\chi} + \mathbf{b}_E [\Delta - \mathbf{v}_{ad} - \mathbf{v}_r] \right] \\ & + \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \Gamma^{(1)-1} \Gamma^{(1)} \cdot \left[\vec{\chi} \cdot \vec{\zeta} \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)T} \right] \right) \\ & + \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \Gamma^{(2)-1} \Gamma^{(2)} \cdot \left[\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) \cdot \vec{\zeta} - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \cdot \mathbf{w}^{(1)} \vec{\chi} \cdot \vec{\zeta} - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)T} \right] \right) \end{aligned} \quad (6.91)$$

Durch Kürzen der Lernraten $\Gamma^{(1)}$ und $\Gamma^{(2)}$, Einsetzen der Gleichung 6.81 sowie der gefilterten

Fehlergröße $\vec{\zeta}$ ergeben sich einige Vereinfachungen:

$$\begin{aligned}\dot{V}(\vec{\chi}) = & \frac{1}{2} \left[\overbrace{\mathbf{A}_E^T \mathbf{P}_E + \mathbf{P}_E \mathbf{A}_E}^{-\mathbf{Q}_E} \right] \cdot \vec{\chi} + \overbrace{\vec{\chi}^T \mathbf{P}_E \mathbf{b}_E}^{\vec{\zeta}} \cdot \left[\Delta - v_{ad} - v_r \right] \\ & + \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \cdot \left[\vec{\chi} \cdot \vec{\zeta} \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)T} \right] \right) \\ & + \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \cdot \left[\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) \cdot \vec{\zeta} - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \cdot \mathbf{w}^{(1)} \vec{\chi} \cdot \vec{\zeta} - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)T} \right] \right). \quad (6.92)\end{aligned}$$

Weiterhin wird die Quantifizierung des Restfehlers $\Delta - v_{ad}$ aus Gleichung 6.75 und der robustifizierende Term aus Gleichung 6.59 eingesetzt:

$$\begin{aligned}\dot{V}(\vec{\chi}) = & -\frac{1}{2} \mathbf{Q}_E \vec{\chi} + \vec{\zeta} \cdot \left[\mathbf{z} + \tilde{\mathbf{w}}^{(2)} \left[\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \mathbf{w}^{(1)} \vec{\chi} \right] + \mathbf{w}^{(2)} \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \tilde{\mathbf{w}}^{(1)} \vec{\chi} \right] \\ & - \vec{\zeta} \cdot \left[k_{r0} + k_{r1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*) \right] \cdot \vec{\zeta}^T \\ & + \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \cdot \left[\vec{\chi} \cdot \vec{\zeta} \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)T} \right] \right) \\ & + \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \cdot \left[\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) \cdot \vec{\zeta} - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \cdot \mathbf{w}^{(1)} \vec{\chi} \cdot \vec{\zeta} - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)T} \right] \right). \quad (6.93)\end{aligned}$$

Nun werden die Terme dieser Gleichung zur deutlicheren Darstellung neu angeordnet.

$$\begin{aligned}\dot{V}(\vec{\chi}) = & -\frac{1}{2} \mathbf{Q}_E \vec{\chi} - \vec{\zeta} \cdot \left[k_{r0} + k_{r1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*) \right] \cdot \vec{\zeta}^T \\ & - \vec{\zeta} \cdot \left[\tilde{\mathbf{w}}^{(2)} \left[\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \mathbf{w}^{(1)} \vec{\chi} \right] + \text{Spur} \left(\left[\tilde{\mathbf{w}}^{(2)} \left[\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \mathbf{w}^{(1)} \vec{\chi} \right] \cdot \vec{\zeta} \right) \right. \right. \\ & \left. \left. - \left[\vec{\zeta} \mathbf{w}^{(2)} \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \right] \cdot \left[\tilde{\mathbf{w}}^{(1)} \vec{\chi} \right] + \text{Spur} \left(\left[\tilde{\mathbf{w}}^{(1)} \vec{\chi} \right] \cdot \left[\vec{\zeta} \mathbf{w}^{(2)} \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \right] \right) \right] \right. \\ & \left. + \vec{\zeta} \cdot \mathbf{z} + \lambda \cdot \|\vec{\zeta}\|_2 \text{Spur} \left(\tilde{\mathbf{w}}^{(2)} \mathbf{w}^{(2)T} \right) + \lambda \cdot \|\vec{\zeta}\|_2 \text{Spur} \left(\tilde{\mathbf{w}}^{(1)} \mathbf{w}^{(1)T} \right) \right). \quad (6.94)\end{aligned}$$

Bedient man sich der Rechenvorschrift $\vec{a}^T \cdot \vec{b} = \text{Spur}(\vec{b} \cdot \vec{a}^T)$, lässt sich eine erhebliche Vereinfachung der Gleichung erreichen, da sich Terme der Lerngesetze mit den entsprechenden der Fehlerdynamik kürzen. Bezieht man außerdem erneut die Zusammenhänge für die Gewichtsfehler aus Gleichung 6.65 ein, erhält man folgende stark vereinfachte Darstellung:

$$\dot{V}(\vec{\chi}) = -\frac{1}{2} \mathbf{Q}_E \vec{\chi} - \vec{\zeta} \left[k_{r0} + k_{r1} \cdot (\|\mathbf{w}\|_F + \bar{\mathbf{w}}_*) \right] \cdot \vec{\zeta}^T + \vec{\zeta} \cdot \mathbf{z} + \lambda \cdot \|\vec{\zeta}\|_2 \cdot \text{Spur}(\tilde{\mathbf{w}} \cdot \mathbf{w}^T). \quad (6.95)$$

Um nun zu bestimmen, unter welchen Umständen $\dot{V}(\vec{\chi}) < 0$ immer gilt, können Schranken für den Fehlervektor $\vec{\chi}$, den gefilterten Fehler $\vec{\zeta}$, den Störterm \mathbf{z} und die Verbindungsgewichte \mathbf{w} in Gleichung 6.95 eingesetzt werden, wie es beispielsweise in [45, 47, 52] und [48] getan wird. In [52] ist die Herleitung aller Schranken und die entsprechende Umformung beschrieben, weshalb an dieser Stelle davon Abstand genommen wird. Gleichwohl soll die Bedeutung von Gleichung 6.95 und ihrer Herleitung noch einmal insofern unterstrichen werden, als dass mit ihrer Hilfe die besonderen Eigenschaften des Gleitzustandslernverfahrens leichter interpretierbar sind.

6.3.3 Erweiterung mit dem Gleitzustandslernverfahren

Das zuvor vorgestellte Verfahren, das den gängigen Rückpropagationsalgorithmus mit den in den Gleichungen 6.78 und 6.79 dargestellten Erweiterungen nutzt, ist für diese Regelungsstrategie Standard [39, 41, 43, 45, 47–49, 51–55]. Obgleich die Leistungsfähigkeit des Ansatzes augenscheinlich gut dokumentiert ist, so verbleibt hierbei doch das Problem, dass die Lernrate μ , respektive die aus ihr zusammengesetzten Matrizen $\Gamma^{(1)}$ und $\Gamma^{(2)}$, empirisch ermittelt oder durch Linearisierung des neuronalen Netzwerkes abgeschätzt werden müssen [43, 52]. Dabei kann es zu den in Abschnitt 2.3 erwähnten Problemen bei zu kleiner oder zu großer Lernrate kommen: Ist μ zu klein, ist die Konvergenzgeschwindigkeit des Verfahrens gering und im geschlossenen Regelkreis können abrupte Störungen, zum Beispiel durch Beschädigungen oder starke Turbulenz, nicht zeitnah abgefangen werden. Ist μ zu groß, werden unter Umständen gute Minima in der Fehlerfläche übersprungen und im Regelkreis können im Sinne der Stabilität zwar begrenzte, aber nichtsdestoweniger unerwünschte Oszillationen der Fehlergröße auftreten. Aus regelungstechnischer Sicht ist also eine dynamische Anpassung der Lernrate wünschenswert, die je nach Flugzustand und äußeren Störungen eine hohe Konvergenzgeschwindigkeit des neuronalen Netzes ermöglicht.

In Kapitel 3 ist das Gleitzustandslernverfahren, das aus der Theorie der strukturvariablen Regelung abgeleitet werden kann, detailliert beschrieben. Es ermöglicht die oben geforderte dynamische Kalkulation der Lernrate bei gleichzeitiger global asymptotischer Stabilität des Verfahrens. Dadurch wird eine erhöhte Konvergenzgeschwindigkeit des Lernalgorithmus erreicht. Außerdem werden die Fehlerzustände des Netzwerkes in den beschriebenen Gleitzustand gezwungen, wobei ein System in demselben sich deutlich unempfindlicher gegenüber äußeren Störungen verhält, was aus regelungstechnischer Sicht nur zu begrüßen ist. Außerdem ist in Kapitel 3 deutlich geworden, dass sich dieses Verfahren rein formal wie ein Rückpropagationsalgorithmus verhält, bei dem lediglich über die Einführung der Schaltfunktion S und ihre Randbedingungen eine dynamische Lernratenbestimmung ermöglicht wird. Dies hat den positiven Effekt, dass die Lernregel aus Gleichung 3.25 direkt in die Lerngesetze für die beiden Gewichtsmatrizen $\mathbf{w}^{(1)}$ sowie $\mathbf{w}^{(2)}$ aus den Gleichungen 6.78 und 6.79 eingesetzt werden kann. Es ergeben sich damit neue und veränderte Gewichts Anpassungen, die den Gleitzustand der Fehlerzustände des KNN garantieren:

$$\dot{\mathbf{w}}^{(1)} = \mu \cdot \mathbf{E} \cdot \left[\text{sign}(S_t) \cdot \vec{x} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \vec{x}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)^T} \right], \quad (6.96)$$

$$\dot{\mathbf{w}}^{(2)} = \mu \cdot \mathbf{E} \cdot \left[\text{sign}(S_t) \cdot \left(\vec{f}(\mathbf{w}^{(1)} \vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)} \vec{x}) \cdot \mathbf{w}^{(1)} \vec{x} \right) \cdot |\vec{\zeta}| - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)^T} \right]. \quad (6.97)$$

Hierbei entspricht der Ausdruck $\mu \cdot \mathbf{E}$ den jeweiligen Matrizen $\Gamma^{(1)}$ und $\Gamma^{(2)}$. Neben der dynamischen Bestimmung der Lernraten und der einfachen Anpassung der Lernregeln gemäß den obigen Gleichungen gibt es weitere erhebliche Vorteile dieses Ansatzes, die im Folgenden eingehender diskutiert werden sollen.

Da die Architektur des neuronalen Netzes mit einer nichtlinearen Transferfunktion in der verdeckten Schicht unangetastet bleibt, ist ersichtlich, dass die Herleitung des verbleibenden Approximationsfehlers des KNN aus den Gleichungen 6.64 bis 6.76 Gültigkeit behält. Gleichzeitig ist in Gleichung 6.97 erkennbar, dass, genau wie zuvor beim klassischen Rückpropagationsalgorithmus, dieser Restfehler durch den zweiten Term im Sinne der Stabilität der Fehlerdynamik kompensiert werden kann. Ebenfalls lässt sich feststellen, dass die zur garantierten Beschränktheit der Verbindungsgewichte eingeführte e-Modifikation wie zuvor in beiden Lernregeln verwendet wird. Dass dieser Ansatz auch bei Nutzung des Gleitzustandslernverfahrens die Verbindungsgewichte begrenzt, soll im weiteren Verlauf noch erörtert werden. Aus diesen Ausführungen und dem Aufbau der Lernregeln aus den Gleichungen 6.96 und 6.97 kann der Schluss gezogen werden, dass die Stabilitätsbetrachtung für ein System mit Rückpropagationsalgorithmus aus Abschnitt 6.3.2 bei begrenzten Fehlern genauso für ein solches Gültigkeit besitzt, das das Gleitzustandslernverfahren nutzt. Dies liegt auch darin begründet, dass über die Definition der Schaltfunktion aus Gleichung 3.20 erreicht werden kann, dass für begrenzte Fehler $\text{sign}(S_l) \cdot |\zeta| = \zeta$ gilt. In diesem Fall sind die Lernregeln der beiden Algorithmen rein formal gleich und können die gleiche Ljapunow-Funktion zur Stabilitätsbetrachtung nutzen.

Bezüglich der Stabilität des geschlossenen Regelkreises existieren weitere Vorteile, die aus der Existenz und dem Erreichen des Gleitzustandes in der Fehlerdynamik des neuronalen Netzwerkes abgeleitet werden können und auch bei beliebig großen Fehlern gelten. Bereits in Kapitel 3 ist dargestellt worden, dass die Fehlerzustände $\vec{\epsilon}$ und $\vec{\epsilon}$ des KNN über die Definition einer geeigneten Schaltlinie S in den Gleitzustand gebracht werden können. Dabei gleiten die Fehlerzustände wegen der intrinsischen Stabilitätsbedingung des Lernverfahrens entlang der Schaltlinie zum stabilen Ursprung des Phasendiagrammes, um dort schließlich zu Null zu werden. Da im diskutierten Fall aber $\vec{\epsilon} = \zeta$ und damit auch $\dot{\vec{\epsilon}} = \dot{\zeta}$ gilt, wird die gefilterte Fehlergröße ζ durch die Stabilitätsforderung des Lernverfahrens in den Gleitzustand gebracht. Gleichzeitig ist sie im Gleitzustand unempfindlicher gegen äußeren Störungen, was bei dem vorgestellten Flugregler zum Beispiel beim Flug in starker Turbulenz von Vorteil ist. Betrachtet man nun noch einmal Gleichung 6.92 als Startpunkt der Stabilitätsanalyse, so sieht man, dass alle Terme der Funktion entweder direkt oder proportional über die Bedingung $\vec{\chi}^T = \zeta \cdot \mathbf{P}_E^{-1} \cdot \mathbf{b}_E^{-1}$ von ζ abhängen. Wird die gefilterte Fehlergröße also in den Gleitzustand gezwungen, so wird damit automatisch sichergestellt, dass die Stabilitätsbedingung aus Gleichung 6.95 erfüllt wird. Die Effektivität der Stabilitätsforderung aus Gleichung 3.23 wird also auch an dieser Stelle für den geschlossenen Regelkreis noch einmal deutlich.

Ein zusätzlicher Vorteil dieser Lernstrategie wird im weiteren Verlauf bei der konkreten Auslegung der einzelnen Regelschleifen für Längs- und Seitenbewegung deutlich werden. Wie zuvor bereits angedeutet, werden jeweils drei kaskadische Schleifen des relativen Grades von Eins verwendet. Dies geschieht, da je nach Architektur bei einem höheren relativen Grad entweder wichtige Größen wie etwa Anstell- und Schiebewinkel wegen der internen Dynamik des Regel-

kreises nicht mehr direkt über die Stellgrößen steuerbar oder höhere Zeitableitungen kommandierter Größen nur schwer zu bestimmen sind. Formuliert man nun die Fehlerdynamik gemäß Gleichung 6.63 für die Rotationsdynamik eines Flugzeuges als System des relativen Grades von Eins, so ergibt sich folgender Zusammenhang:

$$\dot{\vec{e}}_\omega = \mathbf{A}_{E,\omega} \cdot \vec{e}_\omega + (\vec{\Delta}_\omega - \vec{v}_{ad,\omega} - \vec{v}_{r,\omega}) \cdot \mathbf{b}_{E,\omega} . \quad (6.98)$$

An dieser Gleichung lässt sich erkennen, dass $\dot{\vec{\chi}} = \dot{\vec{e}}$ und $\vec{\chi} = \vec{e}$ gelten muss. Überträgt man dies auf die gefilterte Fehlergröße, so wird diese nach der Vorschrift $\vec{\zeta} = \vec{e}_\omega^T \cdot \mathbf{P}_E \cdot \mathbf{b}_E$ bestimmt. Dabei ist \mathbf{b}_E eine Einheitsmatrix, so dass die Vektoren $\vec{\zeta}$ und \vec{e}_ω^T augenscheinlich über die konstante Matrix \mathbf{P}_E ineinander überführt werden können. Ohne der Auslegung der Reglerschleifen im Detail vorwegzugreifen, wird hieran deutlich, dass über die Definition der Schaltfunktion S als Fehlersystem erster Ordnung gemäß Gleichung 3.20 sowie die Abhängigkeit von $\vec{\chi} = \vec{e}$ und $\vec{\zeta}$ auch die Zustände der Fehlerdynamik \vec{e} und ihre zeitliche Änderung $\dot{\vec{e}}$ durch das KNN in einen Gleitzustand gebracht werden. Dass Gleitzustände in der Fehlerdynamik grundsätzlich möglich sind, wird in [120] gezeigt, wo für ein System des relativen Grades von Zwei der Linearregler durch einen Gleitzustandsregler ersetzt wird. Dies stellt eine wichtige Erkenntnis dar, da durch das Gleitzustandslernverfahren wegen seiner variablen Lernrate nicht nur eine verbesserte Konvergenzgeschwindigkeit des Lernverfahrens besonders unter Störungseinfluss realisiert werden kann, sondern tatsächlich auch die zu regelnden Größen der Fehlerdynamik im Gleitzustand stabilisiert werden können. Dieses regelungstechnisch motivierte Trainingsverfahren ermöglicht letztlich also durch die dynamische Kalkulation der Lernraten die Herstellung des stabilen Gleitzustandes, so dass der geschlossene Regelkreis global asymptotisch stabil ist und ohne die näherungsweise oder empirische Bestimmung der Lernraten auskommt.

Wie bereits erwähnt, stellt sich bei der Verwendung dieses Lernansatzes ebenfalls die Frage, ob die Verbindungsgewichte des KNN für sich nach oben beschränkt sind. Um dies zu untersuchen, wird auf die in [52] beschriebene Bestimmung oberer Gewichtsschranken für den Rückpropagationsalgorithmus zurückgegriffen und diese für das Gleitzustandslernverfahren adaptiert. Dazu wird wie zuvor eine Ljapunow-Funktion für das im Regelkreis befindliche Netzwerk gewählt:

$$V(\vec{\chi}) = \frac{1}{2} \text{Spur} \left(\mathbf{w}^{(1)} \Gamma^{(1)-1} \mathbf{w}^{(1)T} \right) + \frac{1}{2} \text{Spur} \left(\mathbf{w}^{(2)} \Gamma^{(2)-1} \mathbf{w}^{(2)T} \right) . \quad (6.99)$$

Da die Bedingung $\dot{V}(\vec{\chi}) < 0$ erfüllt sein muss, wird die Zeitableitung der obigen Gleichung gebildet.

$$\dot{V}(\vec{\chi}) = \text{Spur} \left(\mathbf{w}^{(1)} \Gamma^{(1)-1} \dot{\mathbf{w}}^{(1)T} \right) + \text{Spur} \left(\mathbf{w}^{(2)} \Gamma^{(2)-1} \dot{\mathbf{w}}^{(2)T} \right) \quad (6.100)$$

Hier können nun die Lernregeln aus den Gleichungen 6.96 und 6.97 eingesetzt werden, was den Ausdruck gemäß Gleichung 6.101 ergibt.

$$\begin{aligned} \dot{V}(\vec{\chi}) = & \text{Spur} \left(\mathbf{w}^{(1)} \cdot \left[\text{sign}(S_t) \cdot \vec{\chi} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(1)T} \right] \right) \\ & + \text{Spur} \left(\mathbf{w}^{(2)} \cdot \left[\text{sign}(S_t) \cdot \left(\vec{f}(\mathbf{w}^{(1)} \vec{\chi}) - \mathbf{f}'(\mathbf{w}^{(1)} \vec{\chi}) \cdot \mathbf{w}^{(1)} \vec{\chi} \right) |\vec{\zeta}| - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \mathbf{w}^{(2)T} \right] \right) \end{aligned} \quad (6.101)$$

Diese Gleichung wird zur weiteren Vereinfachung zunächst umgestellt:

$$\begin{aligned}\dot{V}(\vec{\chi}) = & \text{Spur}\left(\mathbf{w}^{(1)} \cdot \left[\text{sign}(S_t) \cdot \vec{x} \cdot |\vec{\zeta}| \cdot \mathbf{w}^{(2)} \cdot \mathbf{f}'(\mathbf{w}^{(1)}\vec{x})\right]\right) \\ & + \text{Spur}\left(\mathbf{w}^{(2)} \cdot \left[\text{sign}(S_t) \cdot \left(\vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot \mathbf{w}^{(1)}\vec{x}\right) \cdot |\vec{\zeta}|\right]\right) \\ & - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \left[\text{Spur}\left(\mathbf{w}^{(2)}\mathbf{w}^{(2)T}\right) + \text{Spur}\left(\mathbf{w}^{(1)}\mathbf{w}^{(1)T}\right)\right].\end{aligned}\quad (6.102)$$

Nun kann erneut die Rechenvorschrift $\vec{a}^T \cdot \vec{b} = \text{Spur}(\vec{b} \cdot \vec{a}^T)$ angewendet werden, was folgenden Zusammenhang liefert:

$$\begin{aligned}\dot{V}(\vec{\chi}) = & |\vec{\zeta}| \cdot \mathbf{w}^{(2)} \cdot \left[\text{sign}(S_t) \cdot \vec{x} \cdot \mathbf{w}^{(1)} \cdot \mathbf{f}'(\mathbf{w}^{(1)}\vec{x})\right] \\ & + |\vec{\zeta}| \cdot \mathbf{w}^{(2)} \cdot \left[\text{sign}(S_t) \cdot \left(\vec{f}(\mathbf{w}^{(1)}\vec{x}) - \mathbf{f}'(\mathbf{w}^{(1)}\vec{x}) \cdot \mathbf{w}^{(1)}\vec{x}\right)\right] \\ & - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \left[\text{Spur}\left(\mathbf{w}^{(2)}\mathbf{w}^{(2)T}\right) + \text{Spur}\left(\mathbf{w}^{(1)}\mathbf{w}^{(1)T}\right)\right].\end{aligned}\quad (6.103)$$

Es ist zu erkennen, dass diese Gleichung durch die zuvor geleistete Umformung erheblich vereinfacht werden kann.

$$\dot{V}(\vec{\chi}) = |\vec{\zeta}| \cdot \mathbf{w}^{(2)} \cdot \text{sign}(S_t) \cdot \vec{f}(\mathbf{w}^{(1)}\vec{x}) - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \left[\text{Spur}\left(\mathbf{w}^{(2)}\mathbf{w}^{(2)T}\right) + \text{Spur}\left(\mathbf{w}^{(1)}\mathbf{w}^{(1)T}\right)\right] \quad (6.104)$$

Unter Verwendung entsprechender Vektor- und Matrixnormen kann nun eine obere Schranke für die Zeitableitung der Ljapunow-Funktion $\dot{V}(\vec{\chi})$ gefunden werden [52]. Dabei kann auch eine obere Schranke für den Ausdruck $\vec{f}(\mathbf{w}^{(1)}\vec{x})$ ermittelt werden, da die Ausgabe der Tangens Hyperbolicus-Funktion asymptotisch begrenzt ist. Zur Bestimmung dieser Schranke wird die Anzahl der Neuronen in der verborgenen Schicht $q^{(2)}$ und der Biaswert von 1 herangezogen:

$$\left\|\vec{f}(\mathbf{w}^{(1)}\vec{x})\right\|_2 = \sqrt{q^{(2)} + 1}. \quad (6.105)$$

Damit lässt sich für die Zeitableitung $\dot{V}(\vec{\chi})$ folgender Zusammenhang finden:

$$\dot{V}(\vec{\chi}) \leq \|\vec{\zeta}\|_2 \cdot \|\mathbf{w}\|_F \cdot \text{sign}(S_t) \cdot \sqrt{q^{(2)} + 1} - \lambda \cdot \|\vec{\zeta}\|_2 \cdot \|\mathbf{w}\|_F^2. \quad (6.106)$$

Nach Umstellen erkennt man, dass die Forderung nach $\dot{V}(\vec{\chi}) \leq 0$ erfüllt werden kann, wenn folgende Ungleichung gilt:

$$\|\mathbf{w}\|_F \geq \frac{\text{sign}(S_t) \cdot \sqrt{q^{(2)} + 1}}{\lambda}. \quad (6.107)$$

Daran ist zu erkennen, dass das Vorzeichen der Schaltfunktion S_t keine Bedeutung für die obere Grenze der Netzwerkgewichte hat und sich schließlich exakt die gleiche Bedingung wie für den Rückpropagationsalgorithmus ergibt [52]. Ebenfalls ist aus Gleichung 6.107 ersichtlich, dass der Faktor λ der e-Modifikation in der Tat Einfluss auf die obere Grenze der Verbindungsgewichte hat.

6.4 Umsetzung der Regelungsstrategie

Nachdem in den vorangegangenen Abschnitten die theoretischen Grundlagen der dynamischen Inversion und die Erweiterungen des Regelkreises um lineare Regelungsanteile, die dynamische Begrenzung der Ersatzregelgrößen sowie neuronale Netze dargestellt worden sind, soll nun die konkrete Umsetzung diskutiert werden. Grundlage der Umsetzung stellt die Entscheidung dar, sowohl Längs- als auch Seitenbewegung als Kaskade dreier Schleifen mit dem relativen Grad von Eins auszuführen, wie es auch in [43, 52] und [54] getan wird. Dabei entspricht die Summe der relativen Grade der Einzelschleifen dem des Gesamtsystems. Es ist genauso möglich, Schleifen eines höheren relativen Grades zu implementieren, allerdings haben die bisherigen Ausführungen gezeigt, dass die zu regelnde Fehlerdynamik des geschlossenen Regelkreises dann auch von entsprechend höheren Ableitungen der Fehlergrößen abhängig ist. Diese Ableitungen in angemessener Qualität messtechnisch verfügbar zu machen, ist nicht immer gegeben und zum Teil auf spezielle Filterarchitekturen angewiesen, wie in [45] dargestellt. Zusätzlich ist die mathematische Umsetzung mit höheren relativen Graden und den dazugehörigen Inversionen ungleich aufwändiger. Weiterhin kommt bei der Nutzung von Schleifen eines höheren relativen Grades unter Umständen das bereits dargestellte Problem zum Tragen, dass bestimmte Zustände des Regelkreises Bestandteil der internen Dynamik werden und damit nicht mehr über die Systemausgänge beobachtbar oder gezielt steuerbar sind. Dies gilt zum Beispiel für die Lageregelung mit einer Schleife des relativen Grades von Zwei und Inversion der Eulerwinkel an Stelle der Anströmwinkel. Diese Problematik ist in [52] anschaulich dargestellt; dabei zeigt sich, dass Anstellwinkel α , Schiebewinkel β und Flugwindhängewinkel μ als Zustände der internen Dynamik ungewolltes Schwingungsverhalten zeigen können, was in Folge die Flugleistungen und -eigenschaften reduziert.

Zusätzlich ist zuvor gezeigt worden, dass sich die Verwendung von Schleifen des relativen Grades von Eins als vorteilhaft für das Gleitzustandslernverfahren darstellt, da die Zustände der Fehlerdynamik und ihre zeitlichen Änderungen durch das Erreichen des Gleitzustandes zusätzlich gegenüber äußeren Störungen stabilisiert und dadurch schneller ausgeglichen werden können. Diese Ausführungen kombiniert mit den zuvor diskutierten Erweiterungen des Reglers führen zu dem in Abbildung 6.8 dargestellten Gesamtsystem. Dabei lässt sich erkennen, dass jede Schleife des Regelkreises durch die Vorsteuerung der Referenzmodelle und potentiell auch durch neuronale Netze beeinflusst werden kann. Während in [54] nur die Rotationsdynamik und in [52] zusätzlich die Bahndynamik durch ein KNN gestützt werden, sollte genau wegen der positiven Effekte des Gleitzustandslernverfahrens eine Implementierung für alle Schleifen grundsätzlich in Erwägung gezogen werden. Der erhöhte Rechenaufwand der zusätzlichen Netzwerke erscheint dabei gegenüber der Leistungssteigerung und zusätzlichen Robustheit gerechtfertigt. Außerdem wird ein Positionsregler eingeführt, der den Spline-basierten Ansatz aus [113] nutzt und in Abschnitt 5.2.1 bereits erläutert wurde. Während die grundsätzliche Anwendbarkeit für unbemannte Flugsysteme zuvor schon angesprochen wurde, ist dieses Verfahren in [105] auch für ein Flugzeug ohne Seitenruder untersucht und optimiert worden.

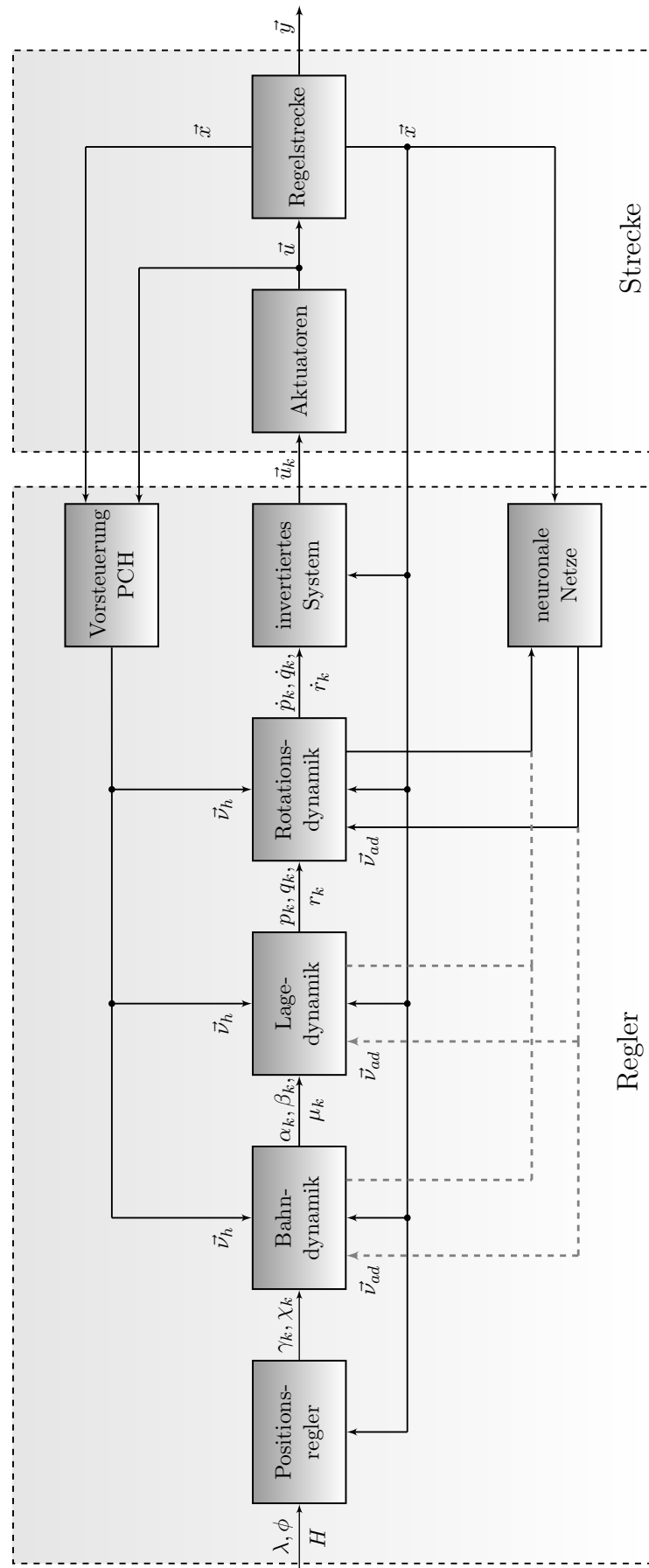


Abbildung 6.8: Übersicht des geschlossenen Regelkreises und aller seiner Erweiterungen.

6.4.1 Anwendung auf die Rotationsdynamik

In Anlehnung an den kaskadischen Aufbau linearer Flugregelungssysteme kann die innerste Schleife als Dämpfer aufgefasst werden [28]. Generell ist bei der Auslegung des gesamten Regelkreises darauf zu achten, dass die Zeitkonstanten der Referenzmodelle ausreichend unterschiedlich sind, um Kopplungseffekte zu vermeiden. Daraus ergibt sich, dass die Rotationsdynamik als innerste Schleife die schnellste und die außen liegende Bahndynamik die langsamste Schleife im geschlossenen System darstellt. Ziel der innersten Schleife ist demnach die Regelung der flugzeugfesten Drehgeschwindigkeiten $\vec{\omega}_f^{gf} = [p, q, r]_f^T$. Grundlage für die Inversion der Rotationsdynamik ist die Differentialgleichung für die Drehgeschwindigkeit, die aus dem Drehimpulssatz abgeleitet wird:

$$\dot{\vec{\omega}}_f^{gf} = \mathbf{T}_f^{-1} \cdot \left[\sum \vec{M}_f - \vec{\omega}_f^{gf} \times (\mathbf{T}_f \cdot \vec{\omega}_f^{gf}) \right]. \quad (6.108)$$

Dabei entspricht \mathbf{T}_f dem Trägheitstensor für ein symmetrisches Flugzeug und $\sum \vec{M}_f$ ist die Summe aller am Flugzeug angreifenden Momente. Da die Rotationsdynamik als System des relativen Grades von Eins betrachtet wird, setzt sich der Vektor der Ersatzregelgrößen aus den Drehbeschleunigungen zusammen:

$$\vec{v}_\omega = \vec{\omega}_f^{gf} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_f. \quad (6.109)$$

Bevor die Eingangs-/Ausgangslinearisierung durchgeführt wird, sei angemerkt, dass in Anlehnung an [43] und [52] ein inkrementeller Ansatz zur Bestimmung des zur Einhaltung der kommandierten Drehbeschleunigungen \vec{v}_ω nötigen Gesamtmomentes gewählt wird. Damit können über die später im Detail zu erläuternde Steuerflächenzuweisung für eine Änderung des Momentenhaushaltes die nötigen Änderungen der Stellgrößen bestimmt werden. Um dies umzusetzen, wird das angreifende Gesamtmoment $\sum \vec{M}_f$ in einen Anteil aufgeteilt, der durch den aktuellen Flugzustand bestimmt ist, und in einen, der die kommandierten Drehbeschleunigungen erzeugt:

$$\sum \vec{M}_f = \underbrace{\left[\vec{M}_0(\vec{x}, \vec{u}) \right]_f}_{\text{aktueller Flugzustand}} + \underbrace{\left[\delta \vec{M}_k(\delta \vec{u}_k) \right]_f}_{\text{erforderliches Inkrement}} = \begin{bmatrix} L_0 \\ M_0 \\ N_0 \end{bmatrix}_f + \begin{bmatrix} \delta L_k \\ \delta M_k \\ \delta N_k \end{bmatrix}_f. \quad (6.110)$$

Dieses Ergebnis kann nun zusammen mit demjenigen aus Gleichung 6.109 in Gleichung 6.108 eingesetzt und invertiert werden, woraus sich das Momenteninkrement in Abhängigkeit der Ersatzregelgröße \vec{v}_ω ergibt.

$$\left(\delta \vec{M}_k(\delta \vec{u}_k) \right)_f = \mathbf{T}_f \cdot \vec{v}_\omega + \vec{\omega}_f^{gf} \times (\mathbf{T}_f \cdot \vec{\omega}_f^{gf}) - \left(\vec{M}_0(\vec{x}, \vec{u}) \right)_f \quad (6.111)$$

Wie bereits zuvor angedeutet, wird die genaue Zuweisung der Steuerflächen separat in Abschnitt 6.4.4 vorgenommen. Da der prinzipielle Ablauf nun verdeutlicht ist, müssen noch die Referenzmodelle und die Ersatzregelgröße \vec{v}_ω genauer erläutert werden. Da diese Zusammenhänge für alle Drehraten die gleichen sind, erfolgt die Notation in allgemeiner Vektorform; die Referenzmodelle erster Ordnung werden folgendermaßen beschrieben:

$$\dot{\vec{\omega}}_R = \frac{1}{T_\omega} \cdot (\vec{\omega}_k - \vec{\omega}_R) - \vec{v}_{\dot{\omega},h} . \quad (6.112)$$

Für die Ersatzregelgröße \vec{v}_ω kann schließlich folgender Zusammenhang gefunden werden:

$$\vec{v}_\omega = \frac{1}{T_\omega} \cdot (\vec{\omega}_k - \vec{\omega}_R) + \mathbf{K}_\omega \cdot (\vec{\omega}_R - \vec{\omega}) . \quad (6.113)$$

Dabei enthält die Diagonalmatrix \mathbf{K}_ω die Verstärkungsfaktoren der Linearregler der Rotationsdynamik und hat die folgende Form:

$$\mathbf{K}_\omega = \begin{bmatrix} k_p & 0 & 0 \\ 0 & k_q & 0 \\ 0 & 0 & k_r \end{bmatrix} . \quad (6.114)$$

Damit sind alle linearen Komponenten der Rotationsdynamik beschrieben, so dass die Erweiterung durch neuronale Netze erfolgen kann. Im Gegensatz zur üblichen Verfahrensweise wird hierbei, wie zuvor im Zusammenhang mit dem Gleitzustandslernverfahren erwähnt, jeder Stellgröße ein KNN zugeordnet. Bei einem Flugzeug, das die Seitenbewegung nur über das Querruder steuert, sind es demnach zwei, bei einer konventionellen Konfiguration mit Seitenruder folglich drei Netzwerke. Die resultierende Form des Regelkreises mit all seinen Komponenten ist in Abbildung 6.9 als Blockschaltbild dargestellt. Zur näheren Beschreibung der Rotationsdynamik wird nun die allgemein gefasste Form ihrer Fehlerdynamik aus Gleichung 6.98 aufgegriffen, wobei die Matrix $\mathbf{b}_{E,\omega}$ eine Einheitsmatrix ist und außerdem $\mathbf{A}_{E,\omega} = -\mathbf{K}_\omega$ gilt.

$$\dot{\vec{e}}_\omega = -\mathbf{K}_\omega \cdot \vec{e}_\omega + \vec{\Delta}_\omega - \vec{v}_{ad,\omega} - \vec{v}_{r,\omega} \quad (6.115)$$

Der Fehlervektor der Rotationsdynamik hat dabei die folgende Form:

$$\vec{e}_\omega = \begin{bmatrix} e_p \\ e_q \\ e_r \end{bmatrix} = \begin{bmatrix} p_R - p \\ q_R - q \\ r_R - r \end{bmatrix} . \quad (6.116)$$

Um nun die gefilterte Fehlergröße $\vec{\zeta}_\omega = \vec{e}_\omega^T \cdot \mathbf{P}_{E,\omega} \cdot \mathbf{b}_{E,\omega}$, also die Trainingssignale der Netzwerke, zu bestimmen, muss folglich die Matrix $\mathbf{P}_{E,\omega}$ ermittelt werden. Dazu wird auf Gleichung 6.81 zurückgegriffen und $\mathbf{A}_{E,\omega} = -\mathbf{K}_\omega$ gesetzt, so dass man unter Verwendung von $\mathbf{Q}_{E,\omega} = 2 \cdot \mathbf{E}$ folgende Lösung erhält:

$$\mathbf{P}_\omega = \begin{bmatrix} 1/k_p & 0 & 0 \\ 0 & 1/k_q & 0 \\ 0 & 0 & 1/k_r \end{bmatrix} . \quad (6.117)$$

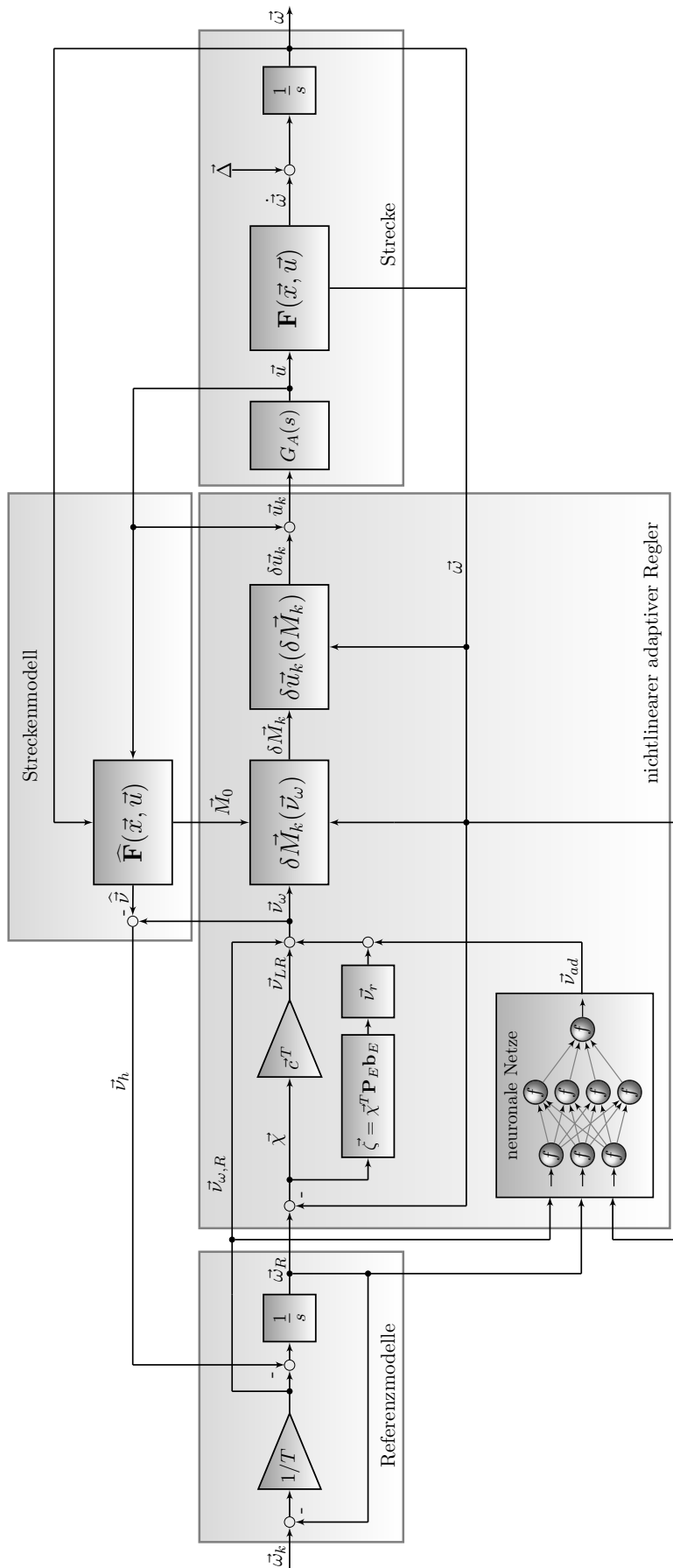


Abbildung 6.9: Anwendung der Regelung auf das Mehrgrößensystem der Rotationsdynamik.

Damit kann schließlich die gefilterte Fehlergröße gemäß Gleichung 6.118 berechnet werden.

$$\vec{\zeta}_\omega = \vec{e}_\omega^T \cdot \mathbf{P}_{E,\omega} \cdot \mathbf{b}_{E,\omega} = \left[\frac{e_p}{k_p}, \frac{e_q}{k_q}, \frac{e_r}{k_r} \right] \quad (6.118)$$

An Gleichung 6.118 ist nun noch einmal deutlich zu erkennen, wie die gefilterte Fehlergröße mit der Fehlerdynamik zusammenhängt und dass die Überführung von $\vec{\zeta}$ und dessen Zeitableitung in einen Gleitzustand den gleichen positiven Effekt auf die entsprechenden Größen der Fehlerdynamik besitzt. Zuletzt bleibt noch, die Zusammensetzung der Eingabevektoren \vec{x}_ω der neuronalen Netze näher zu diskutieren. Allgemein lässt sich hierzu feststellen, dass es sich bei der Approximation des Inversionsfehlers im Grunde um die Abbildung eines dynamischen Prozesses gemäß Abschnitt 2.5 handelt. Bei der Wahl der Eingabegrößen des Netzwerkes können zwar nicht alle in diesem Abschnitt getroffenen Annahmen exakt übertragen werden, gleichwohl sei auf die zeitliche Abhängigkeit dieser Abbildung verwiesen. Man kann berechtigterweise annehmen, dass die Güte der aktuellen Approximation des Inversionsfehlers auch auf den Ausgaben des Netzwerkes der Vergangenheit und den dazugehörigen Zustandsgrößen basiert, weshalb im Gegensatz zu [43, 52] und [54] auch zeitverzögerte Signale verwendet werden. Hinsichtlich der konkreten Auswahl der Netzeingänge ist zu Gleichung 6.62 bereits angemerkt worden, dass neben den Referenz- und Fehlergrößen auch relevante Zustandsgrößen verwendet werden. Darüber hinaus ist es zur Bestimmung einer optimierten Topologie möglich, den in Abschnitt 2.7 vorgestellten Entwurfsprozess für diese Anwendung zu adaptieren. Das Vorgehen hierzu besteht darin, für eine repräsentative Flugenveloppe exemplarische Trainingsdaten in einer Simulation zu generieren, mit denen dann der statistische Trainingsprozess durchgeführt werden kann. Dies ermöglicht schließlich eine fundiertere Auswahl der Netzwerkeingänge aus der Menge der verfügbaren dynamischen Größen des Systems.

Da im weiteren Verlauf, je nach diskutiertem Fluggerät, sowohl zwei als auch drei Netzwerke für die Approximation des Inversionsfehlers in der Rotationsdynamik zum Einsatz kommen, sollen die denkbaren Eingangssignale zunächst nach Längs- und Seitenbewegung aufgeteilt werden. Dabei werden alle jene Eingänge aufgelistet, die auf Grund ihres Informationsgehaltes mit entsprechend zeitverzögerten Werten am Beginn der Topologiedefinition gemäß Abschnitt 2.7 stehen können. Für die Längsbewegung lassen sich unter der Bedingung der Messbarkeit die in Tabelle 6.1 zusammengestellten dynamischen Größen finden.

| Bezeichnung | Symbol | Einheit |
|---------------|-----------|------------------|
| Nickrate | q | rad/s |
| Anstellwinkel | α | ° |
| Nickwinkel | Θ | ° |
| Staudruck | \bar{q} | N/m ² |

Tabelle 6.1: Mögliche Eingangssignale der Längsbewegung.

Der Staudruck \bar{q} wird als Eingangsgröße herangezogen, da er den Einfluss der Flugeschwindigkeit auf die Umströmung des Flugzeuges charakterisiert und somit eine wichtige Messgröße darstellt. Daher ist es sinnvoll, diesen auch als Eingangssignal für die Seitenbewegung in Betracht zu ziehen, für die die möglichen Eingangsgrößen in Tabelle 6.2 zusammengestellt sind.

| Bezeichnung | Symbol | Einheit |
|---------------|-----------|------------------|
| Rollrate | p | rad/s |
| Gierrate | r | rad/s |
| Schiebewinkel | β | ° |
| Hängewinkel | Φ | ° |
| Staudruck | \bar{q} | N/m ² |

Tabelle 6.2: Mögliche Eingangssignale der Seitenbewegung.

Es sei außerdem angemerkt, dass auch bei einer Aufteilung auf mehrere neuronale Netze eine Berücksichtigung von Kopplungseffekten zwischen Längs- und Seitenbewegung erfolgen kann. Dazu können in Frage kommende Drehraten und besonders Winkel aus den Tabellen 6.1 und 6.2 einem Netzwerk der jeweils anderen Bewegungsform als Eingänge hinzugefügt werden.

6.4.2 Anwendung auf die Lageregelung

Die Aufgabe der Lageregelungsschleife besteht darin, aus den kommandierten Größen der weiter außen in der Kaskade liegenden Bahnregelung kommandierte Drehraten für die Rotationsdynamik zu bestimmen. Dafür sind, wie zuvor schon angedeutet, zwei Ansätze möglich, nämlich die Nutzung der Eulerwinkel Φ , Θ , Ψ oder die der Bahnwinkel α_K , β_K und μ_K . Während die Eulerwinkel über ein Lagereferenzsystem gut messbar sind und sie sich diesbezüglich folglich anbieten, besitzt ihre Nutzung vor allem den zuvor erwähnten Nachteil von Zuständen in der internen Dynamik des geschlossenen Regelkreises. Dieses Problem kann mit der Nutzung der Bahnwinkel umgangen werden, wobei sich hier, besonders bei kleinen unbemannten Flugsystemen, die berechnete Frage ihrer Messbarkeit stellt. Unter anderem in [121] wird gezeigt, dass mit geeigneter Sensorik auch für kleine Fluggeräte eine zuverlässige Messung von Anstell- und Schiebewinkel möglich ist. Daher sollen im Hinblick auf die interne Dynamik die Winkel α_K , β_K und μ_K verwendet werden. Die folgende Herleitung der relevanten Gleichungen orientiert sich an [54], wobei ihre regelungstechnische Umsetzbarkeit auch in [105] gezeigt ist. Zunächst wird der Vektor der Ersatzregelgrößen für die Lagedynamik gemäß Gleichung 6.119 definiert.

$$\vec{v}_{Lage} = \begin{bmatrix} v_{\dot{\alpha}} \\ v_{\dot{\beta}} \\ v_{\dot{\mu}} \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_K \\ \dot{\beta}_K \\ \dot{\mu}_K \end{bmatrix} \quad (6.119)$$

Betrachtet man das Flugzeug als Punktmasse und geht nicht von der Vereinfachung der Windstille aus, so kann man gemäß [54] die benötigten Winkelableitungen schreiben zu:

$$\dot{\alpha}_K = q_K - \frac{q_A}{\cos \beta_K} - p_K \cdot \cos \alpha_K \cdot \tan \beta_K - r_K \cdot \sin \alpha_K \cdot \tan \beta_K, \quad (6.120)$$

$$\dot{\beta}_K = r_A + p_K \cdot \sin \alpha_K - r_K \cdot \cos \alpha_K, \quad (6.121)$$

$$\dot{\mu}_K = p_A + (q_A \cdot \sin \mu_K + r_K \cdot \cos \mu_K) \cdot \tan \gamma. \quad (6.122)$$

Dabei können die im aerodynamischen Koordinatensystem definierten Drehraten folgendermaßen bestimmt werden:

$$p_A = p_K \cdot \cos \alpha_K \cdot \cos \beta_K + (q_K - \dot{\alpha}_K) \cdot \sin \beta_K + r_K \cdot \sin \alpha_K \cdot \cos \beta_K, \quad (6.123)$$

$$q_A = -\frac{F_{A,z}}{m \cdot V_K} - \frac{g}{V_K} \cdot \cos \gamma \cdot \cos \mu_K, \quad (6.124)$$

$$r_A = \frac{F_{A,y}}{m \cdot V_K} + \frac{g}{V_K} \cdot \cos \gamma \cdot \sin \mu_K. \quad (6.125)$$

Zur Berechnung der Gleichungen 6.124 und 6.125 müssen die Komponenten des im aerodynamischen Koordinatensystem definierten Kraftvektors F_A bestimmt werden. Ein Teil von F_A besteht aus den Triebwerkskräften F_{TW} , welche folglich in das aerodynamische System transformiert werden müssen. Die Transformationsmatrix M_{af} wird dabei gemäß den Vorschriften aus [94] berechnet, so dass für den Kraftvektor F_A folglich gilt:

$$F_A = \begin{bmatrix} -W \\ Q \\ -A \end{bmatrix}_a + \mathbf{M}_{af} \cdot \begin{bmatrix} F_{TW,x} \\ F_{TW,y} \\ F_{TW,z} \end{bmatrix}_f. \quad (6.126)$$

Um unter diesen Rahmenbedingungen die Lösbarkeit der Gleichungen 6.120 bis 6.122 zu ermöglichen, werden zusätzlich folgende Vereinfachungen angenommen: Zum einen sei der Schubeinstellwinkel gleich Null, die Triebwerkskraft wirke also nur in x_f -Richtung; zum anderen soll davon ausgegangen werden, dass der Schiebewinkel β durch die Regelung klein gehalten wird und demnach die üblichen Näherungen $\sin \beta \approx 0$ und $\cos \beta \approx 1$ genutzt werden können. An dieser Stelle sei angemerkt, dass eine solche Näherung nicht mehr ohne weiteres angenommen werden darf, wenn der Schiebewinkel ein Zustand der internen Dynamik wäre. Damit können nun die Gleichungen 6.123 bis 6.126 in die Gleichungen 6.120 bis 6.122 eingesetzt werden.

$$\dot{\alpha}_K = q_K - \frac{A + F_{TW} \cdot \sin \alpha}{m \cdot V_K} + \frac{g}{V_K} \cdot \cos \gamma \cdot \cos \mu_K \quad (6.127)$$

$$\dot{\beta}_K = p_K \cdot \sin \alpha_K - r_K \cdot \cos \alpha_K - \frac{Q}{m \cdot V_K} + \frac{g}{V_K} \cdot \cos \gamma \cdot \sin \mu_K \quad (6.128)$$

$$\dot{\mu}_K = p_K \cdot \cos \alpha_K + p_K \cdot \sin \alpha_K + \frac{\tan \gamma}{m \cdot V_K} \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_K - Q \cdot \cos \mu_K \right] \quad (6.129)$$

Aus diesen Gleichungen ist ersichtlich, dass die auftretenden Kräfte A , Q und F_{TW} bestimmt werden müssen. Dies geschieht in dem bereits in Abbildung 6.9 dargestellten Flugzeugmodell,

das parallel zur Regelung betrieben wird und auch die nötigen Berechnungen zur Vorsteuerung der Referenzmodelle durchführt. Löst man die Gleichungen 6.127 bis 6.129 nun nach den flugzeugfesten Drehraten auf, setzt die Beziehungen aus Gleichung 6.119 ein und indiziert wie zuvor alle kommandierten Größen mit k , so erhält man folgende Zusammenhänge für die Inversion der Lagedynamik:

$$p_{K,k} = v_{\dot{\mu}} \cdot \cos \alpha_{K,k} + v_{\dot{\beta}} \cdot \sin \alpha_{K,k} + \frac{Q \cdot \sin \alpha_{K,k}}{m \cdot V_{K,k}} - \frac{g}{V_{K,k}} \cdot \cos \gamma_k \cdot \sin \mu_{K,k} \cdot \sin \alpha_{K,k} - \frac{\tan \gamma_k \cdot \cos \alpha_{K,k}}{m \cdot V_{K,k}} \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_{K,k} + Q \cdot \cos \mu_{K,k} \right], \quad (6.130)$$

$$q_{K,k} = v_{\dot{\alpha}} + \frac{1}{m \cdot V_{K,k}} \cdot (A + F_{TW} \cdot \sin \alpha) - \frac{g}{V_{K,k}} \cdot \cos \gamma_k \cdot \cos \mu_{K,k}, \quad (6.131)$$

$$r_{K,k} = v_{\dot{\mu}} \cdot \sin \alpha_{K,k} - v_{\dot{\beta}} \cdot \cos \alpha_{K,k} + \frac{Q \cdot \cos \alpha_{K,k}}{m \cdot V_{K,k}} + \frac{g}{V_{K,k}} \cdot \cos \gamma_k \cdot \sin \mu_{K,k} \cdot \sin \alpha_{K,k} - \frac{\tan \gamma_k \cdot \cos \alpha_{K,k}}{m \cdot V_{K,k}} \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_{K,k} + Q \cdot \cos \mu_{K,k} \right]. \quad (6.132)$$

Wie zuvor in Abschnitt 6.4.1 können nach der Darstellung der Inversion nun die Referenzmodelle und die Bestimmung der Ersatzregelgröße \vec{v}_{Lage} erläutert werden. Dabei soll an dieser Stelle im Sinne der Übersichtlichkeit das Vorgehen am Beispiel des Anstellwinkels dargestellt werden. Für alle drei Winkel sind die verwendeten Referenzmodelle erneut Systeme erster Ordnung, was anhand Gleichung 6.133 deutlich wird.

$$\dot{\alpha}_{K,R} = \frac{1}{T_{\alpha}} \cdot (\alpha_{K,k} - \alpha_{K,R}) - v_{\dot{\alpha},h} \quad (6.133)$$

Die Ersatzregelgröße $v_{\dot{\alpha}}$ kann mit folgender Beziehung ermittelt werden, wobei angemerkt sei, dass das zusätzliche Integralglied, repräsentiert durch den letzten Term auf der rechten Seite, optional in seiner Nutzung ist und auch ein reines P-Glied denkbar wäre:

$$v_{\dot{\alpha}} = \frac{1}{T_{\alpha}} \cdot (\alpha_{K,k} - \alpha_{K,R}) + K_{\alpha,P} \cdot (\alpha_{K,R} - \alpha_K) + K_{\alpha,I} \cdot \int (\alpha_{K,R} - \alpha_K) \cdot dt. \quad (6.134)$$

Da es sich beim Anstellwinkel wegen der Möglichkeit des Strömungsabrisses um eine empfindliche Größe handelt, sei darauf hingewiesen, dass es bei der Ermittlung der Referenzverläufe zweckmäßig ist, sowohl den Anstellwinkel selbst als auch seine Änderungsrate $\dot{\alpha}$ zu begrenzen. Dies kann durch entsprechende Beschränkungen innerhalb des Referenzmodells realisiert werden, wobei im Falle der Änderungsrate auch eine dynamische Begrenzung denkbar ist. Es ist sinnvoll, dies zu tun, da eine erst einmal aufgebaute Rate nicht abrupt zurückgefahren werden kann, wenn ein Maximalanstellwinkel α_{max} erreicht ist [52]. Das um die erforderlichen Begrenzungen erweiterte Referenzmodell ist in Abbildung 6.10 sichtbar. Dabei ist auch eine Begrenzung hinter dem Integrator erkennbar, welche sicherstellt, dass das Vorsteuersignal $v_{\dot{\alpha},h}$ nicht zu unzulässigen Referenzanstellwinkeln führt.

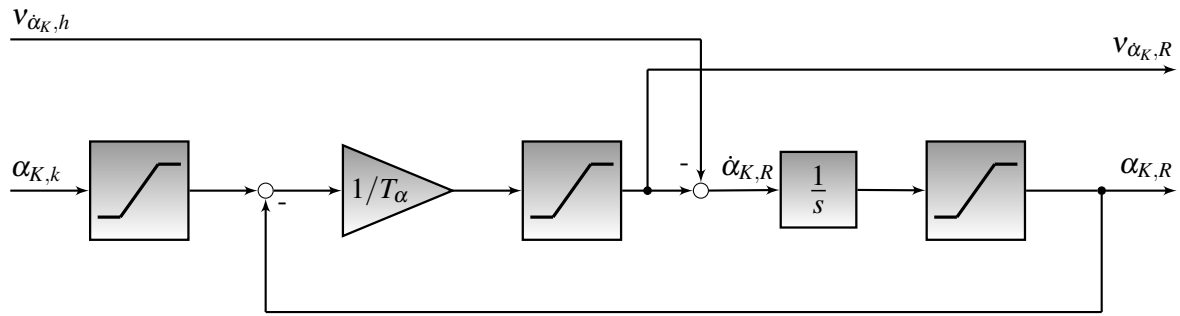


Abbildung 6.10: Um Begrenzungen erweitertes Referenzmodell erster Ordnung.

In [52] und [54] ist dargestellt, dass eine Erweiterung der Lageregelungsschleife mit neuronalen Netzen bei der Verwendung des Rückpropagationsalgorithmus' als Lernverfahren kaum Vorteile bietet. Wie in Abschnitt 6.3.3 erläutert, ändert sich dies allerdings bei Nutzung des Gleitzustandslernverfahrens, da ein solches KNN die Zustände der Fehlerdynamik im Gleitzustand gegen externe Störungen stabilisiert und so die Robustheit des geschlossenen Regelkreises erhöht. Die potentielle Erweiterung dieser Regelschleife folgt dabei dem gleichen Schema, wie es im vorigen Abschnitt für die Rotationsdynamik erläutert ist und soll an dieser Stelle daher nicht weiter vertieft werden.

6.4.3 Anwendung auf die Bahndynamik und Bahnvorgabe

Die äußerste Schleife der vorgestellten Reglerkaskade stellt die Bahndynamik dar. Neben der Generierung der Triebwerksstellgröße ist es ihre maßgebliche Aufgabe, aus den Bahngrößen der Geschwindigkeit V_K , des Flugbahnwinkels γ und Kurswinkels χ die kommandierten Größen $\alpha_{K,k}$, $\beta_{K,k}$ und $\mu_{K,k}$ der Lageregelungsschleife zu bestimmen. Der zugehörige Vektor der Ersatzregelgrößen \vec{v}_{Bahn} ist wie folgt definiert:

$$\vec{v}_{Bahn} = \begin{bmatrix} v_{\dot{V}_K} \\ v_{\dot{\gamma}} \\ v_{\dot{\chi}} \end{bmatrix} = \begin{bmatrix} \dot{V}_K \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix}. \quad (6.135)$$

Um nun die kommandierten Winkel für die Lageregelung bestimmen zu können, wird die Inversion der Bahndynamik durchgeführt wie sie auch in [54] beschrieben ist. Ausgegangen wird hierbei erneut von der Betrachtung des Flugzeuges als Punktmasse, womit sich folgende Gleichungen formulieren lassen:

$$\dot{V}_K = \frac{F_{A,x}}{m} - g \cdot \sin \gamma, \quad (6.136)$$

$$\dot{\gamma} = q_A \cdot \cos \mu_K - r_A \cdot \cos \mu_K, \quad (6.137)$$

$$\dot{\chi} = \frac{1}{\cos \gamma} \cdot (q_A \cdot \sin \mu_K + r_A \cdot \sin \mu_K). \quad (6.138)$$

Bereits im vorigen Abschnitt sind in den Gleichungen 6.123 bis 6.126 sowohl die Drehraten im aerodynamischen Koordinatensystem, als auch die am Flugzeug angreifenden Kräfte beschrieben worden. Nutzt man weiterhin dieselben Vereinfachungen wie zuvor hinsichtlich kleiner Winkel und Schubrichtung, kann man die obigen Gleichungen wie folgt umformen:

$$\dot{V}_K = \frac{F_{TW} \cdot \cos \alpha - W}{m} - g \cdot \sin \gamma, \quad (6.139)$$

$$\dot{\gamma} = \frac{1}{m \cdot V_K} \cdot \left[(A + F_{TW} \cdot \sin \alpha) \cdot \cos \mu_K + Q \cdot \sin \mu_K \right] - \frac{g}{V_K} \cdot \cos \gamma, \quad (6.140)$$

$$\dot{\chi} = \frac{1}{m \cdot V_K \cdot \cos \gamma} \cdot \left[(A + F_{TW} \cdot \sin \alpha) \cdot \sin \mu_K - Q \cdot \cos \mu_K \right]. \quad (6.141)$$

Um dieses Gleichungssystem nun nach den gesuchten Größen F_{TW} , A und μ_K umzustellen, wird im Sinne einer eindeutigen Lösung eine zusätzliche Vereinfachung nötig. Es wird angenommen, dass durch die Regelung der aerodynamische Schiebewinkel sehr klein ist, wodurch die Querkraft Q gegenüber den anderen Anteilen vernachlässigt werden kann. Setzt man die Beziehungen aus Gleichung 6.135 ein und indiziert wie zuvor die kommandierten Größen mit k , so führt die Umformung zu:

$$F_{TW,k} = \frac{m(\dot{V}_K + g \cdot \sin \gamma_k) + W}{\cos \alpha_K}, \quad (6.142)$$

$$A_k = m \cdot \cos \gamma_k \cdot \sqrt{(V_{K,k} \cdot \dot{\chi})^2 + \left(\frac{V_{K,k} \cdot \dot{\gamma}}{\cos \gamma_k} + g \right)^2} - \left[m(\dot{V}_K + g \sin \gamma_k) + W \right] \cdot \tan \alpha_K, \quad (6.143)$$

$$\mu_{K,k} = \arctan \left(\frac{V_{K,k} \cdot \dot{\chi} \cdot \cos \gamma_k}{V_{K,k} \cdot \dot{\gamma} + g \cdot \cos \gamma_k} \right). \quad (6.144)$$

Während sich der kommandierte Bahnhängewinkel $\mu_{K,k}$ direkt ergibt, sind für die Bestimmung der anderen Größen weitere Schritte erforderlich. Die Schubhebelstellung δ_F kann unter Zuhilfenahme eines zuvor ermittelten Schubkennfeldes aus $F_{TW,k}$ bestimmt werden. Aus dem kommandierten Auftrieb A_k lässt sich weiterhin ein kommandierter Auftriebsbeiwert $C_{A,k}$ ermitteln, so dass durch Inversion des Kennfeldes $C_A(\alpha)$ schließlich das Kommando für den Anstellwinkel abzuleiten ist. Um weiterhin einen schiebefreien Flug zu gewährleisten, wird der Bahnschiebewinkel $\beta_{K,k}$ stets entgegen eines gemessenen Windwinkels kommandiert. Wie zuvor soll im weiteren noch auf die Referenzmodelle und die Ersatzregelgrößen der Bahndynamik eingegangen werden. Da die Referenzmodelle erneut für alle Größen den gleichen Aufbau als System erster Ordnung aufweisen, sollen sie am Beispiel des Bahnwinkels γ beschrieben werden. Der entsprechende Zusammenhang ist in Gleichung 6.145 ersichtlich.

$$\dot{\gamma}_R = \left[\frac{1}{T_\gamma} \cdot (\gamma_k - \gamma_R) \right]_{\lim} - \dot{\gamma}_{\gamma,h} \quad (6.145)$$

Für die Bestimmung der Ersatzregelgröße gilt folgender Zusammenhang, wobei auch hier für jede Größe eine zusätzliche Erweiterung durch Integralanteile denkbar ist:

$$v_{\dot{\gamma}} = \left[\frac{1}{T_{\gamma}} \cdot (\gamma_k - \gamma_R) \right]_{\lim} + K_{\gamma} \cdot (\gamma_R - \gamma) . \quad (6.146)$$

An den beiden obigen Gleichungen ist ersichtlich, dass zur Einhaltung existierender Flugbereichsgrenzen in allen Referenzmodellen Begrenzungen implementiert sind. Der Bahnwinkel und die Fluggeschwindigkeit werden in Abhängigkeit des Staudruckes als Maß für die Geschwindigkeit vorgegeben. Dabei stellen die Referenzmodelle sicher, dass die entsprechenden Kommandos nicht zu einer Fluggeschwindigkeit unterhalb der Minimalgeschwindigkeit und damit zu einem Strömungsabriss führen. Ebenso werden die erlaubte maximale Geschwindigkeit und Beschleunigung nach oben begrenzt. Die maximale Kursänderungsrate $\dot{\chi}$ lässt sich ebenfalls durch geeignete Maßnahmen im Referenzmodell begrenzen, wie es in [52] gezeigt wird. Dabei betrachtet man für den jeweiligen Flugzustand die geometrischen Beziehungen der Kräfte in Vertikal- und Horizontalrichtung am Flugzeug. Unter Berücksichtigung des verfügbaren Maximalauftriebes kann eine Grenze für das maximale aerodynamische Lastvielfache $n_{z,aero,max}$ gefunden werden, das durch eine kommandierte Kursänderungsrate und die damit verbundenen zusätzlichen Kräften nicht überschritten werden darf. Der Wert für das Lastvielfache kann dabei ebenfalls mit einem konstruktiv bedingten Maximalwert kombiniert werden, um strukturelle Schäden am Flugzeug zu vermeiden. Mit den enthaltenen Begrenzungen sind die Referenzmodelle der Bahndynamik folglich dem des Anstellwinkels aus Abbildung 6.10 vom Aufbau her ähnlich.

Die Erweiterung dieser Regelschleife durch neuronale Netze kann analog zu dem bereits beschriebenen Ansatz durchgeführt werden. Dabei ist es zweckmäßig, die Eingabesignale der Netze um solche Messgrößen zu erweitern, die das Verhalten der Bahndynamik näher beschreiben. Sofern messtechnisch verfügbar, können der Bahnazimut χ , der Bahnwinkel γ , die aktuelle Geschwindigkeit V_K , die Höhe über Grund H oder auch die Höhenänderung \dot{H} hierfür verwendet werden. Die für den vollautomatischen Flug notwendigen kommandierten Bahngrößen $V_{K,k}$, γ_k , χ_k oder auch H_k und deren Regelabweichungen können ebenfalls in Betracht gezogen werden. Allerdings ist dabei stets darauf zu achten, dass die Signale selbst eine hinreichende Dynamik aufweisen, da zu gleichförmige Eingangssignale wegen mangelnder Unterscheidungsmöglichkeit einzelner Trainingsmuster den Lernprozess eines KNN behindern oder gar verzerren können.

Die bereits angesprochenen kommandierten Größen zur Einhaltung einer gewünschten Flugbahn werden in der in Abbildung 6.8 dargestellten Positionsregelung generiert. Zum Einsatz kommt hierfür die zuvor in Abschnitt 5.2.1 vorgestellte Bahnführung auf Spline-Kurven. Der Aspekt, dass diese Kurven überall zweifach stetig differenzierbar sind, stellt dabei sicher, dass die als Kommandogrößen zu berechnenden ersten Ableitungen bis in die innerste Regelschleife keine unerwünschten Sprünge aufweisen. Eine Flugmission besteht in der Regel aus einer Vielzahl einzelner, miteinander verbundener Kurvenstücke, wird meist im Vorfeld erstellt und als Flugplan im Regelungssystem hinterlegt. Im Rahmen des Regelungsprozesses stellt die Positi-

onsregelung dabei nach der Bestimmung der Flugzeugposition im Raum und relativ zur Bahn in jedem Zeitschritt folgende Größen zur Verfügung: Die im Flugplan definierte kommandierte Geschwindigkeit $V_{K,k}$, die kommandierte Flughöhe H_k , die kommandierte Höhenänderung \dot{H}_k , die seitliche Bahnablage d und einen aus der Geometrie der Bahn und der Position des Flugzeuges ermittelten vorgesteuerten Bahnazimut χ_{vor} [113]. Während $V_{K,k}$ also direkt aus dem Flugplan vorgegeben wird, werden für γ_k und χ_k Zwischenschritte benötigt. Unter erneuter Betrachtung des Flugzeuges als Punktmasse und Vernachlässigung der Erdkrümmung kann für den kommandierten Bahnwinkel folgender Zusammenhang gefunden werden:

$$\gamma_k = \arctan \frac{\dot{H}_k}{\sqrt{\dot{x}_k^2 + \dot{y}_k^2}} . \quad (6.147)$$

Für die Berechnung des kommandierten Bahnazimuts χ_k werden die zwei Signale des Positionsreglers χ_{vor} und d mit einer einfachen Proportionalschaltung miteinander gemäß Gleichung 6.148 kombiniert.

$$\chi_k = \chi_{vor} + K_{P,d} \cdot d \quad (6.148)$$

Durch diese Kombination werden sowohl Informationen über die Geometrie der gewünschten Flugbahn, als auch die aktuelle Relativposition zur kommandierten Bahn in die Bestimmung von χ_k einbezogen. Es sei zusätzlich angemerkt, dass das Konzept der Bahnführung und -planung bei der Nutzung dieser Regelungsstrategie für bemannte Flugzeuge im Sinne einer generischen Verwendung für lange Flugstrecken entsprechend zu erweitern wäre. Für kleinere unbemannte Systeme hat sich dieser Ansatz hingegen in einer Vielzahl wissenschaftlicher Flugmissionen als in der Praxis gut anwendbar erwiesen [115, 121].

6.4.4 Steuerflächenzuweisung

Wie bereits in Abschnitt 6.4.1 erläutert, wird durch die Reglerkaskade ein Vektor kommandierter Momenteninkremente $(\delta \vec{M}_k)_f$ bestimmt, die es über eine geeignete Zuordnung verfügbarer Stellgrößen zu realisieren gilt. Dabei kann man leicht nachvollziehen, dass diese Zuordnung deutlich komplexer ausfällt, wenn neben Quer-, Höhen- und Seitenruder noch, wie bei Verkehrsflugzeugen üblich, verschiedene Klappen und Spoiler mitbetrachtet werden müssen. Ähnliches gilt, wenn beim Ausfall eines Stellorgans der Momentenhaushalt im Rahmen einer Rekonfiguration durch ein anderes gesteuert werden soll. Dazu sei angemerkt, dass das vorgestellte Regelungssystem wegen seiner Adaptivität gut für Rekonfigurationsuntersuchungen auch bei bemannten Flugsystemen geeignet ist [50]. Ebenso ist die Kombination mit lernfähigen Fehlerdetektionsverfahren möglich, die neuronale Netzwerke verwenden [122], und beispielsweise auch das Gleitzustandslernverfahren zur robusten Fehlererkennung nutzen können [83]. Da derlei Aspekte nicht den Schwerpunkt dieser Arbeit ausmachen, soll für das betrachtete Regelungsverfahren von einer Konfiguration mit Höhenruder η , Querruder ξ und Seitenruder ζ ausgegangen werden.

Um nun die Zusatzausschläge der einzelnen Ruder bestimmen zu können, lassen sich aus dem Vektor der Momenteninkremente $(\delta \vec{M}_k)_f$ zunächst Änderungen der dimensionslosen Momentenbeiwerte berechnen:

$$\delta C_l = \frac{\delta L_k}{\bar{q} \cdot S \cdot b/2}, \quad \delta C_m = \frac{\delta M_k}{\bar{q} \cdot S \cdot l_\mu}, \quad \delta C_n = \frac{\delta N_k}{\bar{q} \cdot S \cdot b/2}. \quad (6.149)$$

Die Berechnung der dafür erforderlichen zusätzlichen Ruderausschläge kann unter Kenntnis der beteiligten aerodynamischen Derivative gemäß Gleichung 6.150 erfolgen.

$$\begin{bmatrix} \delta C_l \\ \delta C_m \\ \delta C_n \end{bmatrix} = \begin{bmatrix} 0 & C_{l\xi} & C_{l\zeta} \\ C_{m\eta} & 0 & 0 \\ 0 & C_{n\xi} & C_{n\zeta} \end{bmatrix} \cdot \begin{bmatrix} \delta \eta \\ \delta \xi \\ \delta \zeta \end{bmatrix} \quad (6.150)$$

Sollte, wie bei kleineren unbemannten Systemen nicht unüblich, auf ein Seitenruder verzichtet werden, so vereinfacht sich obige Gleichung entsprechend dem wegfallenden Seitenruderausschlag ζ . Durch Umstellen von Gleichung 6.150 können schließlich die Zusatzausschläge bestimmt werden, wobei dieses Vorgehen der in [52] vorgestellten näherungsweisen Inversion unter Verwendung von Kennfelddaten entspricht. Zieht man nun noch die aktuellen Stellausschläge hinzu, so können die kommandierten Ruderausschläge schließlich bestimmt werden:

$$\eta_k = \eta_0 + \delta \eta, \quad \xi_k = \xi_0 + \delta \xi, \quad \zeta_k = \zeta_0 + \delta \zeta. \quad (6.151)$$

Hierbei können die aktuellen Ausschläge η_0 , ξ_0 und ζ_0 entweder durch Messung oder über ein identifiziertes Aktuatormodell bestimmt und so dem Regelungsprozess zur Verfügung gestellt werden.

6.5 Parameterwahl für das Gesamtsystem

In den Abschnitten 6.1 bis 6.4 ist das Regelungskonzept mit dynamischer Inversion und den dazugehörigen Erweiterungen vorgestellt worden, so dass an dieser Stelle die Wahl der einzelnen Systemparameter kurz erläutert werden soll. Zu diesen Parametern zählen die Zeitkonstanten der Referenzmodelle, die Verstärkungsfaktoren der Linearregler der einzelnen Schleifen sowie die Parameter des robustifizierenden Terms der KNN. Bei Verwendung des Rückpropagationsalgorithmus' müssen darüber hinaus die Lernraten, zum Beispiel durch Linearisierung des neuronalen Netzes [43, 52], bestimmt werden, was beim Gleitzustandslernverfahren entfällt. Bei der Auslegung der Parameter der linearen Regelungsanteile ist neben der Forderung der Stabilität ebenso eine Optimierung der Bandbreite wünschenswert mit dem Ziel einer möglichst hohen Frequenz, bei der der Amplitudenabfall von Ausgangs- zu Eingangssignal größer als 50 % ist. Als Stabilitätsmaß dient das bereits in Abschnitt 5.2.4 erläuterte Hurwitz-Kriterium. Da das Prinzip der Stabilitätsbetrachtung und der Parameteroptimierung in [52, 54] und [105] detailliert dargestellt ist, sollen hier die wesentlichen Punkte diskutiert werden.

Es wird für jede Regelschleife die zuvor erwähnte Dynamik des linearen Ersatzsystems betrachtet; dabei wird davon ausgegangen, dass durch die Verwendung neuronaler Netzwerke kein Inversionsfehler besteht und die Eingangs-/Ausgangslinearisierung somit ideal ist. Unter diesen Voraussetzungen kann gezeigt werden, dass die Übertragungsfunktion von kommandierter Größe y_k zur Ausgangsgröße y einer jeden Regelschleife s von der Zeitkonstante T_s seines Referenzmodells und der Dynamik der nächst inneren Schleife $s - 1$ abhängt. Im Bildbereich lässt sich dies allgemein wie folgt darstellen:

$$\frac{y(s)}{y_k(s)} = \frac{G_{s-1}(s)}{T_s \cdot s + G_{s-1}(s)} . \quad (6.152)$$

Betrachtet man nun die innerste Schleife der Rotationsdynamik, so muss für deren Übertragungsfunktion die Aktuatordynamik $G_A(s)$ berücksichtigt werden. Diese wird, wie in Abschnitt 6.2 angedeutet, als lineares System zweiter Ordnung betrachtet und weist demzufolge eine Form gemäß Gleichung 5.19 auf. Überträgt man dies auf Gleichung 6.152, so erhält man mit der Zeitkonstante der Rotationsdynamik T_ω folgenden Ausdruck:

$$G_\omega(s) = \frac{G_A(s)}{T_\omega \cdot s + G_A(s)} = \frac{\omega_0^2}{T_\omega \cdot s^3 + 2 \cdot T_\omega D \omega_0 \cdot s^2 + T_\omega \omega_0^2 \cdot s + \omega_0^2} . \quad (6.153)$$

Dies kann nun unter Verwendung der Zeitkonstante T_L für die Lageschleife fortgeführt werden.

$$G_L(s) = \frac{\omega_0^2}{T_\omega T_L \cdot s^4 + 2 \cdot T_\omega T_L D \omega_0 \cdot s^3 + T_\omega T_L \omega_0^2 \cdot s^2 + T_L \omega_0^2 \cdot s + \omega_0^2} \quad (6.154)$$

Analog ergibt sich schließlich für die Bahndynamik die Gesamtübertragungsfunktion des linearen Ersatzsystems.

$$G_B(s) = \frac{\omega_0^2}{T_\omega T_L T_B \cdot s^5 + 2 \cdot T_\omega T_L T_B D \omega_0 \cdot s^4 + T_\omega T_L T_B \omega_0^2 \cdot s^3 + T_L T_B \omega_0^2 \cdot s^2 + T_B \omega_0^2 \cdot s + \omega_0^2} \quad (6.155)$$

Mit diesen Zusammenhängen können die einzelnen Regelschleifen von innen nach außen ausgelegt werden, wobei man sich zur Stabilitätsbetrachtung Hurwitz-Matrizen entsprechend der charakteristischen Polynome der Übertragungsfunktionen zu Nutze macht; für die Rotationsdynamik heißt dies zum Beispiel:

$$\mathbf{H}_\omega = \begin{bmatrix} a_1 & a_3 & 0 \\ a_0 & a_2 & 0 \\ 0 & a_1 & a_3 \end{bmatrix} . \quad (6.156)$$

Führt man diese Stabilitätsbetrachtung durch, so lassen sich Forderungen für die einzelnen Zeitkonstanten der Referenzmodelle ableiten. Die Zeitkonstante der Rotationsdynamik ist demnach

abhängig von Dämpfung und Eigenfrequenz des Aktuators, womit sich folgende Beschränkung finden lässt:

$$T_\omega > T_{\omega,min} = \frac{1}{2D\omega_0} . \quad (6.157)$$

Wird dies bis zur äußersten Schleife fortgesetzt, so können allgemeine Grenzen für die Zeitkonstanten angegeben werden:

$$\frac{T_\omega}{T_{\omega,min}} > 1 , \quad \frac{T_L}{T_\omega} > 1 , \quad \frac{T_B}{T_L} > 1 . \quad (6.158)$$

Man sieht, dass die Zeitkonstanten von innen nach außen zunehmen, dies also zu einer schnellen inneren Schleife und einer deutlich langsameren äußeren Schleife führt. Diese Randbedingungen können weiterhin im Rahmen einer klassischen Regleroptimierung mit einem Gütefunktional gekoppelt werden, wobei neben der erwähnten Bandbreitenoptimierung weitere Einflüsse, wie zum Beispiel eine Mindestdämpfung der Pole des linearen Ersatzsystems, berücksichtigt werden können. In [105] ist ein solches Vorgehen unter Verwendung eines Simplex-Verfahrens detailliert beschrieben. Nach der Optimierung der Zeitkonstanten sind die Verstärkungsfaktoren k_i der einzelnen Linearregler auszulegen. Wegen ihres Einflusses auf die Fehlerdynamik, was zum Beispiel an Gleichung 6.115 erkennbar ist, orientiert sich ihre Wahl an den Zeitkonstanten der jeweiligen Referenzmodelle. Damit die Fehlerdynamik weder deutlich schneller noch langsamer als die optimierte Referenzdynamik ist, wird in [52] und [54] folgende Wahl vorgeschlagen:

$$k_i = \frac{1}{T_i} . \quad (6.159)$$

Eine Orientierung der Geschwindigkeit der Fehlerdynamik an jener der Referenzdynamik erscheint sehr zweckmäßig. Gleichwohl ist im Rahmen von Flugeigenschaftsuntersuchungen auch eine gewisse Veränderung der Parameter um die in Gleichung 6.159 gegebenen Startwerte denkbar. Diese Parameter finden schließlich auch Eingang in die Bestimmung der gefilterten Fehlergröße der neuronalen Netze, was in Gleichung 6.117 bereits dargestellt ist. Zusätzlich sind noch die Parameter des robustifizierenden Terms aus Gleichung 6.59 zu bestimmen. In [52] ist in Anlehnung an [43] für den Faktor k_{r1} im Sinne der Stabilität der Fehlerdynamik folgende untere Grenze angegeben:

$$k_{r1} > \frac{1}{2} \cdot \sqrt{q^{(2)} \cdot \bar{\mathbf{w}}_*} \cdot k_i . \quad (6.160)$$

Der Faktor k_{r0} ist ein optionaler Parameter des robustifizierenden Terms, der beispielsweise in [39] und [43] keine Verwendung findet. Dieser kann als zusätzliches Stabilitätsmaß interpretiert werden, dessen Bestimmung nach [52] allerdings näherungsweise durchgeführt wird. Dazu wird, genau wie zur Bestimmung der Lernraten des Rückpropagationsalgorithmus', das neuronale Netz um einen Idealzustand ohne Inversionsfehler linearisiert [43] und die resultierende Fehlerdynamik vereinfacht als lineares System zweiter Ordnung aufgefasst. Es wird angenommen, dass man die Dämpfung D_{NN} und Frequenz ω_{NN} dieses adaptiven Systems über die

Lernraten des neuronalen Netzes und den Parameter k_{r0} des robustifizierenden Terms vorgeben kann, was für letzteren zu folgendem Ausdruck führt:

$$k_{r0} = k_i \cdot (2 \cdot D_{NN} \cdot \omega_{NN} - k_i) . \quad (6.161)$$

Die Vorgehensweise der Linearisierung des KNN unter der Annahme, dass kein Inversionsfehler existiert, ist ohne Zweifel ein vereinfachender Sonderfall, der für die Bestimmung einer unteren Grenze von k_{r0} durchaus zweckmäßig erscheint. Bei der Bestimmung elementarer Trainingsparameter wie den Lernraten allerdings ist die Berechnung über die intrinsische Stabilitätsforderung des Gleitzustandslernverfahrens dem obigen Ansatz klar vorzuziehen.

Zuletzt ist die Wahl des Parameters λ der e-Modifikation zu untersuchen. Der Term hat, wie zuvor gezeigt, Einfluss auf die oberen Grenzen der Gewichtsmatrizen. Nach [119] soll damit sichergestellt werden, dass die Netzwerkgewichte bei fehlender Anregung des KNN durch einen signifikanten Ausgabefehler nicht langsam ins Unendliche wachsen. Ein stetig marginaler Ausgabefehler hätte folglich eine stetige marginale Gewichtsveränderung zur Folge. Allerdings ist, je nach Lernproblem, dabei nicht gewährleistet, dass die Gewichte nicht einfach dauerhaft in eine Richtung wachsen, um den Fehler zu minimieren. Der über λ definierte *Term des Vergessens* stellt für solche Fälle sicher, dass die Gewichtsmatrizen langsam abgebaut werden. In [39] wird beispielsweise für die Anwendung der Regelung eines Industrieroboters die Wahl von $\lambda = 0,1$ vorgeschlagen. Für die Flugregelungsanwendung sollte man diesen Faktor tendenziell kleiner wählen, da das Verhalten eines Flugzeuges nicht zuletzt durch die atmosphärischen Störungen deutlich dynamischer ausfällt. Ebenso ist eine exakte Systemidentifizierung bei einem Flugzeug ungleich aufwändiger als bei einem Industrieroboter, was dazu führt, dass ein neuronales Netz im Normalfall existierende Fehlerzustände/Inversionsfehler auszugleichen hat und das Problem der zu geringen Anregung und daraus folgend möglicher unendlicher Gewichte unwahrscheinlich ist. Ein zu hohes Maß des Vergessens würde demzufolge der schnellen Anpassung an die jeweilige Flugzeugdynamik entgegenwirken.

7 Ausgewählte Simulationsergebnisse

In diesem Kapitel sollen die vorgestellten Regelungsverfahren und Lernmethoden anhand von Simulationsergebnissen bewertet werden. Diese Betrachtung erfolgt in zwei Teilen; zunächst werden die modellfreie neuronale Regelung und die dynamische Inversion gestützt mit neuronalen Netzen am Beispiel des kleineren Fluggerätes Carolo *P200* miteinander verglichen. Dazu wird eine exemplarische wissenschaftliche Flugmission mit einer Flugdauer von 15 Minuten gewählt, um die Langzeitstabilität beider Regelungsansätze darzulegen. Anschließend erfolgt am Beispiel des Carolo *P360* die Betrachtung der dynamischen Inversion im Nominalfall und ein Vergleich des Rückpropagationsalgorithmus' mit dem Gleitzustandslernverfahren in Gegenwart starker Parameterunsicherheiten, herbeigeführt etwa von Beschädigungen. Sofern nicht anders erwähnt, erfolgen alle Simulationen unter der Verwendung des in Kapitel 4 beschriebenen Störgrößenmodells bei Wind mit überlagerter Turbulenz.

Die beiden genutzten Flugsysteme unterscheiden sich stark in ihren dynamischen Eigenschaften, was sowohl mit Größe und Gewicht als auch mit der unterschiedlichen aerodynamischen Konfiguration begründet ist. Zusätzlich verfügt der *P200* im Sinne einfacher Bauweise zur Steuerung der Seitenbewegung lediglich über Querruder, während der *P360* auch Seitenruder besitzt. Die wichtigsten technischen Daten beider Fluggeräte sind in Tabelle 7.1 zusammengefasst. Die aerodynamischen Beiwerte beider Flugzeuge, die für die flugdynamische Simulation

| | <i>P200</i> | <i>P360</i> |
|--------------------------|-------------|-------------|
| Startgewicht | 5 kg | 25 kg |
| Spannweite | 200 cm | 360 cm |
| Länge | 120 cm | 150 cm |
| Reisefluggeschwindigkeit | 20 m/s | 23 m/s |
| Antrieb | elektrisch | elektrisch |
| max. Nutzlast | 1 kg | 5 kg |

Tabelle 7.1: Technische Daten der unbemannten Flugsysteme Carolo *P200* und *P360*.

und die Regelung mittels dynamischer Inversion benötigt werden, sind in [83] im Rahmen umfangreicher Strömungssimulationen ermittelt und validiert worden. Dadurch ist kombiniert mit den implementierten Sensor- und Aktuatoremodellen eine Simulation des dynamischen Verhaltens mit angemessenem Realitätsgrad gewährleistet.

Da die Abminderung externer Störeinflüsse eine der Kernaufgaben eines Regelungssystems ist, soll vor der Analyse der eigentlichen Reglereigenschaften eine Bewertung des Einflusses des Störgrößenmodells erfolgen. Dies ist nicht zuletzt deswegen relevant, da kleine Flugsysteme deutlich anfälliger gegenüber Böen und Turbulenz sind, als dies für Fluggeräte deutlich höherer Masse der Fall ist. Dazu sind in Abbildung 7.1 die Komponenten des Windvektors u_{Wg} , v_{Wg} und w_{Wg} mit einem überlagerten Turbulenzspektrum exemplarisch dargestellt. Hierbei wird für alle Komponenten ein konstanter Windanteil von 1 m/s mit zufällig generierter Turbulenz kombiniert, was zu dem in Kapitel 4 erwähnten stochastischen Verhalten führt. Es ist erkennbar, dass sich der Betrag der Windkomponenten innerhalb weniger Sekunden um bis zu 5 m/s ändern kann. Bedenkt man weiterhin, dass die Reisefluggeschwindigkeit des P200 bei 20 m/s liegt, lässt sich feststellen, dass die modellierten atmosphärischen Störungen einen erheblichen Einfluss auf die Flugverhalten haben. Das Störgrößenmodell ermöglicht somit, das Reglerverhalten unter anspruchsvollen atmosphärischen Randbedingungen zu untersuchen.

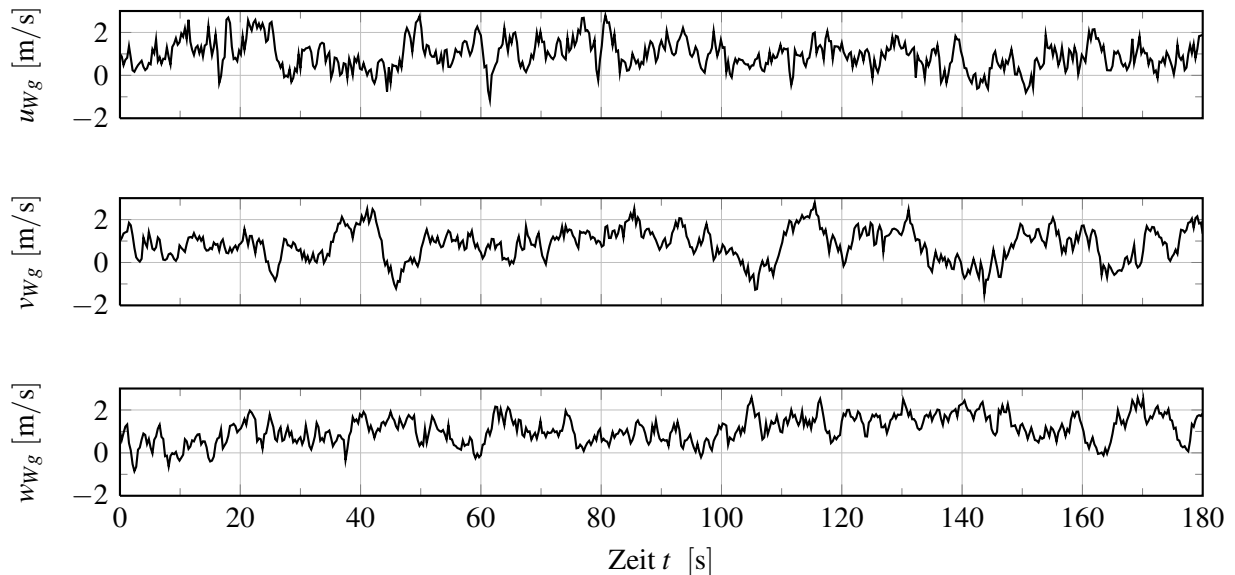


Abbildung 7.1: Komponenten des Windvektors überlagert mit einem Turbulenzspektrum.

7.1 Vergleich der Regelungsansätze

Bei der Regelung unbemannter Flugsysteme besteht eine zentrale Aufgabe im möglichst genauen Einhalten der vorgegebenen Flugmission, auch bei widrigen Windverhältnissen. Sollen zum Beispiel Luftbilder eines bestimmten Areals aufgenommen werden, ist es im Sinne der kompletten Abdeckung des Gebietes mit Bildmaterial und der Datenauswertung wünschenswert, dass die Abweichung von der Zielbahn dauerhaft bei wenigen Metern liegt. Ebenso ist es für jedwede Form der Datengewinnung im Flug von Vorteil, wenn die Fluglage möglichst stabil ist. Beide Randbedingungen gelten besonders auch bei automatischen Starts und Landungen.

Um nun zunächst die Eigenschaften der modellfreien, rein neuronalen Flugregelung zu beurteilen, wird die Regelung der Seitenbewegung des Carolo P200 für den vollautomatischen Flug längerer Missionen herangezogen. Bei der nichtlinearen Simulation des Fluggerätes handelt es sich gemäß Kapitel 4 um eine verkoppelte Bewegung mit sechs Freiheitsgraden, wobei die Regelung der Längsbewegung mittels dynamischer Inversion erfolgt. Unter Berücksichtigung des beschriebenen Störgrößenmodells lässt sich feststellen, dass die Komplexität der betrachteten Regelungsaufgabe höher ist als bei vergleichbaren Untersuchungen, da etwa in [15, 17] und [64] lediglich unverkoppelte nichtlineare Systeme geringerer Ordnung und mit begrenzten Störungen betrachtet werden. Dies gilt auch für die in [63] vorgestellte Flugregelungsanwendung, da hier nur die entkoppelte translatorische Seitenbewegung mit drei Freiheitsgraden, ohne atmosphärische Störungen und für einzelne Manöver betrachtet wird.

Grundlage für die Untersuchung der Regelungseigenschaften ist das Basiswissen von neuronalem Regler und Beobachter, deren Trainingsphase in Abschnitt 5.4 bewertet wurde. Damit die beiden Netzwerke zusammen als nichtlineares, arbeitspunktunabhängiges Regelungssystem nutzbar sind, muss das Vorwissen während der Einsatzphase verallgemeinernd und robust abgerufen werden, während gleichzeitig eine Anpassung an bisher unbekannte Flugsituationen erfolgt. Um dies zu bewerten, wird die Flugmission, die zur Generierung der Trainingsdaten genutzt wurde und in Abbildung 5.9 dargestellt ist, herangezogen und mit einer Abfolge von den Netzwerken unbekannten Manövern auf 300 Sekunden verlängert. Diese Manöver stellen ebenfalls eine Kombination aus Kurven- sowie Steig- und Sinkflügen dar, so dass sich die Flugmission gemäß Abbildung 7.2 ergibt.

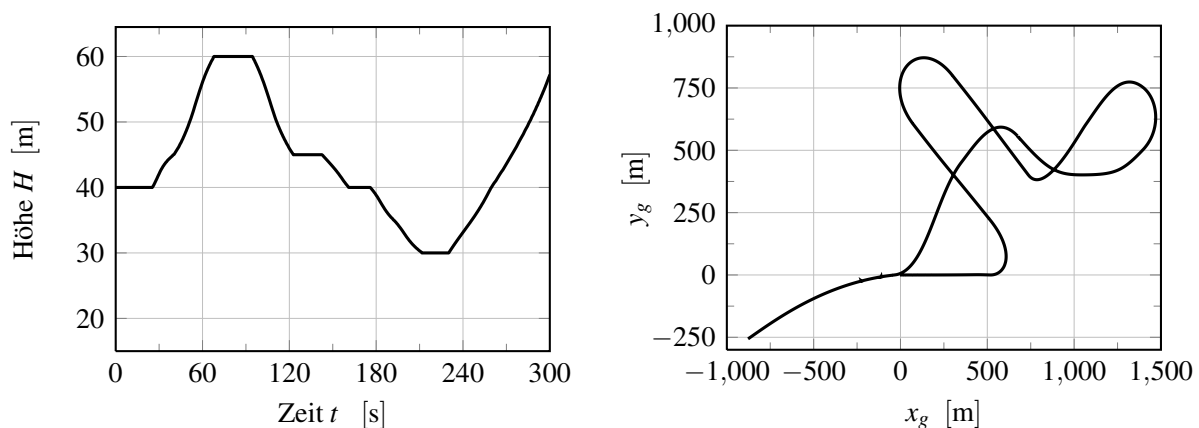


Abbildung 7.2: Nicht maßstäbliches Profil der Testflugbahn in Längs- und Seitenbewegung.

Wird dieses Flugprofil mit Windstörungen ähnlich denen aus Abbildung 7.1 durchfliegen, erzielt die rein neuronale Reglerstruktur unter Verwendung des Gleitzustandslernverfahrens eine seitliche Bahnfolge wie sie in Abbildung 7.3 erkennbar ist. Es wird deutlich, dass, von vereinzelt Abweichungen von mehr als fünf Metern bei anspruchsvolleren Manövern wie dem Abfangen bei gleichzeitigem Kurvenflug abgesehen, die seitliche Bahnablage durchgehend deutlich unter fünf Metern liegt. Das leichte Schwingungsverhalten um die Zielbahn ist dabei als keine all-

zu große Einschränkung der Regelungsqualität anzusehen, bedenkt man beispielsweise, dass in den ersten 25 Sekunden des Fluges auf einer Wegstrecke von 500 Metern lediglich drei Schwingungen mit einer Amplitude von maximal 2,5 Metern auftreten. Ebenfalls zu unterstreichen ist, dass es auf den letzten 120 Sekunden der Mission, die nicht im Vorwissen der Netzwerke enthalten ist, zu keiner erkennbaren Verschlechterung der Regelungseigenschaften kommt. Dies belegt zum einen, dass der Informationsgehalt der Trainingsdaten des Vorwissens angemessen ist und zum anderen, dass die durch geringe Test- und Validierungsfehler konstatierten Generalisierungseigenschaften der Netze in robuster Weise vorhanden sind.

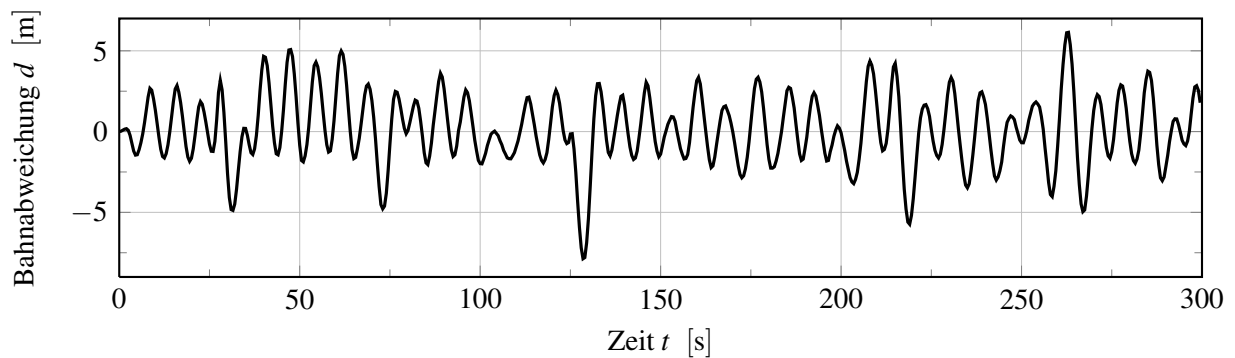


Abbildung 7.3: Seitliche Bahnabweichung einer unbekannten Bahn unter Störeinfluss.

Dass die vorgestellte neuronale Regelungsstrategie die komplexen nichtlinearen Zusammenhänge angemessen abzubilden vermag, wird auch deutlich, wenn man die Ausgabe des Beobachternetzwerkes heranzieht. Der neuronale Beobachter bildet die Teile der Systemdynamik ab, die für die Bestimmung des Lernsignals des Reglernetzwerkes genutzt werden. Als Bestandteil des Kritikelementes ist seine stabile Funktion somit von großer Bedeutung für den Lernprozess des Reglernetzwerkes. In Abbildung 7.4 sind für die betrachtete Flugbahn die Beobachteraussage und der dazugehörige Ausgabefehler dargestellt. Sehr gut zu erkennen ist,

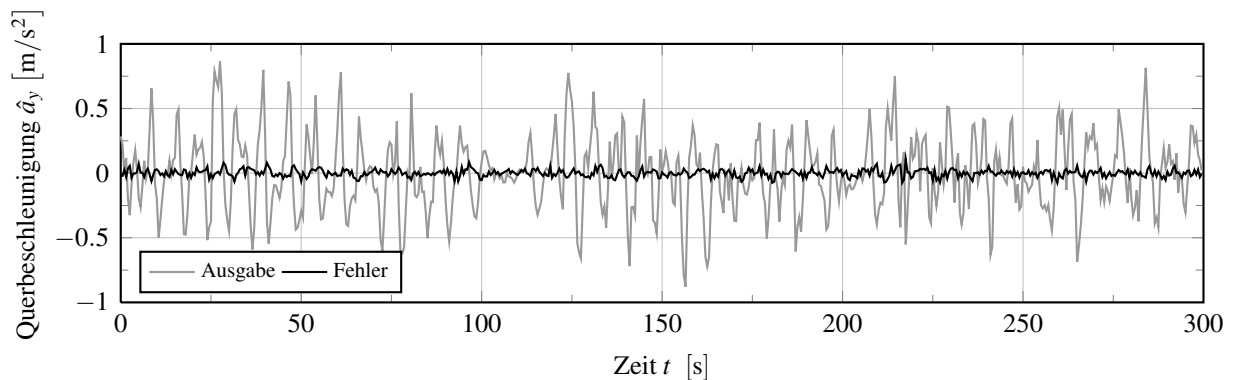


Abbildung 7.4: Ausgabe und Ausgabefehler des neuronalen Beobachters.

dass der Schätzfehler sich trotz der Dynamik der Flugbahn, die sich auch im Ausgabesignal des Beobachters niederschlägt, durchgehend um Null herum befindet. Auch dieses Ergebnis unterstreicht die adäquaten und robusten Abbildungseigenschaften der verwendeten Netzwerke und belegt ebenso das gute Konvergenzverhalten des Gleitzustandslernverfahrens.

Um darüber hinaus die Regelungseigenschaften auch bei stärkeren Störeinflüssen zu analysieren, wird dieselbe Flugbahn bei deutlich höherem Wind durchflogen. Dazu wird die Windkomponente u_{wg} auf 5 m/s gesetzt, so dass es in Kombination mit dem Turbulenzspektrum zu Windgeschwindigkeiten von über 7 m/s und damit zu nachhaltigen Störungen kommen kann. Der resultierende Bahnverlauf der neuronalen Regelung ist in Abbildung 7.5 erkennbar. Es wird deutlich, dass es an einigen Stellen zwar zu höheren Bahnablagen von ca. 10 Metern kommt, wie zum Beispiel bei ungefähr 30 und 130 Sekunden, sich die sonstige Bahnführungsgenauigkeit jedoch nicht signifikant verschlechtert. Die höheren Abweichungen treten dann auf, wenn das Fluggerät durch starken Wind in seiner Kurvenkoordination beeinträchtigt wird und durch hartes Gegensteuern Überschwinger in der Bahnabweichung produziert. Diese Ergebnisse zeigen, dass sich das Regelungssystem auch an stärkere Störungen anzupassen vermag, ohne zu große Einbußen in der Bahnfolge hinnehmen zu müssen.

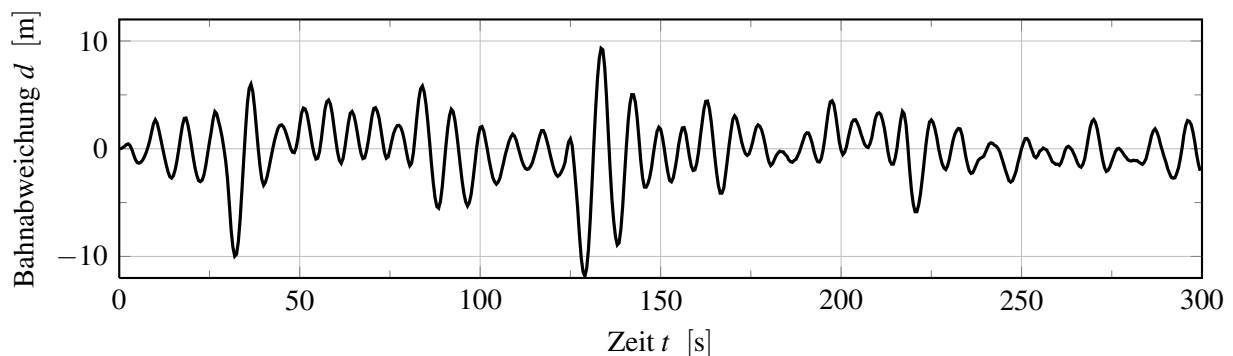


Abbildung 7.5: Seitliche Bahnabweichung bei erhöhtem Wind mit $u_{wg} = 5$ m/s.

Gerade bei stärkeren externen Störungen kann darüber hinaus ein Effekt auftreten, der im geregelten System und der Reglerarchitektur begründet liegt. Da die Regelung der seitlichen Bahnabweichung direkt über das Querruder erfolgt, ist es ähnlich wie in Abschnitt 6.1.3 beschrieben möglich, dass Zustände einer internen Dynamik, die folglich nicht direkt über das Ausgangssignal beobachtbar sind, zu Schwingungen angeregt werden. In Abbildung 7.6 ist dargestellt, dass dies in der Seitenbewegung unter ungünstigen Bedingungen für den Hängewinkel Φ möglich ist. Es ist eine langsame und stabile, aber ausgeprägte und unerwünschte Schwingung des Hängewinkels erkennbar. Sofern angeregt, kann dieser Effekt auftreten, da die Dynamik des Hängewinkels bei der gegebenen Reglerkonfiguration nicht in das Lernsignal des neuronalen Reglers einfließt und unerwünschte Effekte daher nicht direkt unterdrückt werden können.

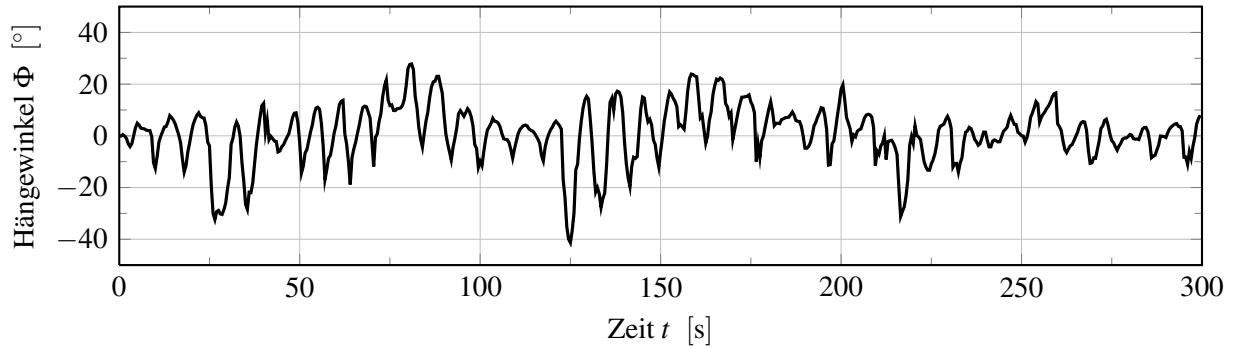


Abbildung 7.6: Durch Störungen angeregte Schwingung des Hängewinkels.

Trotz des möglichen Auftretens begrenzter Schwingungen in der internen Dynamik liefert das rein neuronale Regelungssystem in Anbetracht der Komplexität der Regelungsaufgabe eine insgesamt angemessene Bahnführungsgenauigkeit. Besonders für unbemannte Forschungsflugzeuge stellt sich die Frage, ob das vorgestellte Verfahren auch eine wissenschaftliche Mission im automatischen Flug adäquat zu regeln vermag. Dazu ist in Abbildung 7.7 nun ein exemplarisches Flugmuster etwa zur Gewinnung von Luftbildern qualitativ dargestellt. Hierbei beginnt das Flugzeug die Mission in einer Flughöhe von 40 Metern in x_g -Richtung fliegend und steigt in einer Kreisbahn dann auf die Sollhöhe von 150 Metern, auf der das spiralförmige Missionsprofil durchflogen wird. Das so abgedeckte Areal hat eine Abmessung von etwa einem Quadratkilometer, was einer typischen Mission zur Gewinnung von Luftbildern von kleineren Arealen entspricht. Zum Ende der Mission wird in einer weiteren Kreisbahn ein Sinkflug durchgeführt, so dass das Fluggerät in etwa wieder zur Starthöhe zurückkehrt.

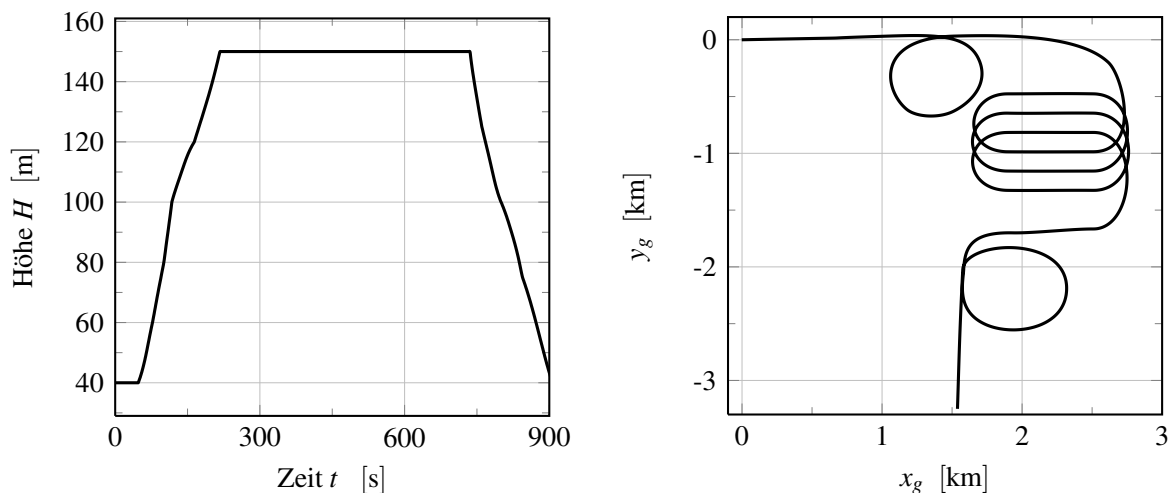


Abbildung 7.7: Profil einer wissenschaftlichen Flugmission in Längs- und Seitenbewegung.

Diese Flugbahn stellt für die rein neuronale Regelung eine anspruchsvolle Aufgabe dar, da sich hieran besonders zeigt, wie gut die Generalisierungseigenschaften der implementierten Netz-

werke ausgeprägt sind und ob der weitere Lernprozess der komplexen nichtlinearen Zusammenhänge im Flug langzeitstabil erfolgt. Dabei ist es auch von Interesse, ob bei den längeren Geradeausflügen eventuell eine Überanpassung an diesen Flugzustand erfolgt, der dann zu reduzierten Regeleigenschaften im Kurvenflug führt. Die entsprechenden Ergebnisse der lateralen Bahnfolge sind in Abbildung 7.8 sichtbar. Man erkennt, dass trotz des Windeinflusses die Bahnablage durchgehend deutlich unter fünf Metern liegt, meist ist sie sogar nicht höher als 2,5 Meter. Lediglich gegen Ende der Mission ist ein leichter, aber unkritischer Anstieg der Abweichung zu verzeichnen, welcher auf das dauerhafte Einwirken des Seitenwindes während des Fluges in negativer y_g -Richtung zurückzuführen ist. Ebenso ist bei zunehmender Zeit keine Verschlechterung des Regelverhaltens zu beobachten, was unterstreicht, dass es im Betrieb nicht zum Verlernen bestimmter Eigenschaften kommt. Die Ergebnisse für diese Flugbahn belegen, dass neuronale Netzwerke nicht nur dynamische Prozesse adäquat abzubilden vermögen, sondern auch komplexe, mit Kopplungseffekten behaftete nichtlineare Regelungsaufgaben erlernen können. Gleichzeitig lässt sich feststellen, dass die Regelung eines Flugzeuges möglich ist, ohne dass überhaupt Modellkenntnisse des Systems erforderlich wären. Dies ist nicht zuletzt für kleine unbemannte Systeme interessant, da sich hier begrenzte Entwicklungsbudgets häufig in ungenauer Parameterkenntnis und damit sehr konservativer Reglerauslegung niederschlagen.

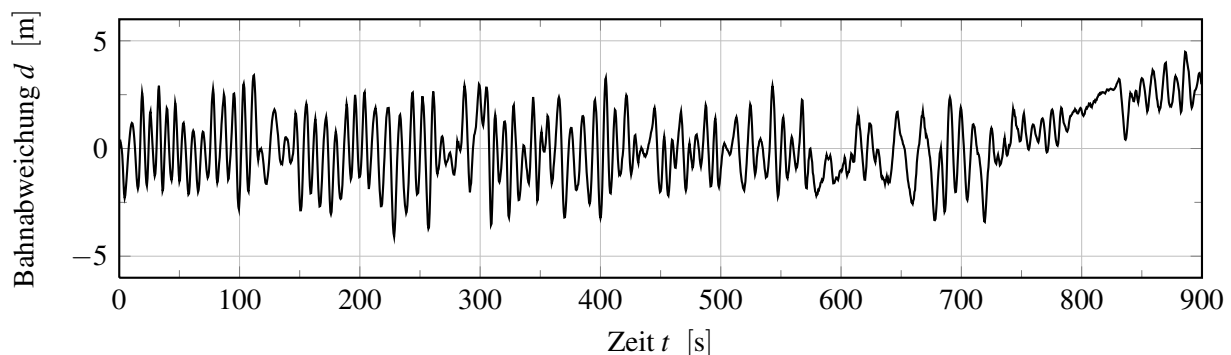


Abbildung 7.8: Seitliche Bahnabweichung für die exemplarische Flugmission.

Zum Vergleich der Regelgüte werden die gleichen Flugbahnen nun unter Verwendung der mit neuronalen Netzwerken erweiterten dynamischen Inversion als Regelungssystem durchflogen. Dabei wird ähnlich wie in [52] und [54] zunächst vereinfachend angenommen, dass das regelungstechnische Modell dem tatsächlichen Flugzeug entspricht und dass die auftretenden Windstörungen exakt gemessen werden können. Die Ergebnisse der lateralen Bahnfolge für die Flugmission aus Abbildung 7.2 sind für die beiden zuvor diskutierten Windstörungen in Abbildung 7.9 dargestellt. Für beide Fälle ist erkennbar, dass die Zielbahn trotz des Windeinflusses sehr genau eingehalten wird. Lediglich in scharfen Kurven oder nach einem im Kurvenflug durchzuführenden Abfangmanöver kommt es wie etwa bei 120 Sekunden zu höheren Abweichungen. Die leichten Schwingungen um die Zielbahn, die bei der modellfreien neuronalen Regelung zu beobachten sind, treten hier nicht auf.

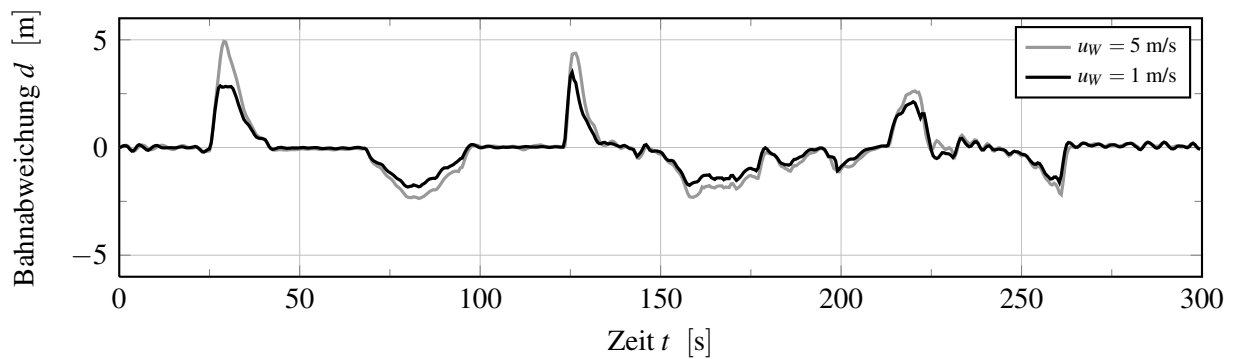


Abbildung 7.9: Seitliche Bahnabweichung bei unterschiedlichen Windverhältnissen.

Die guten Regelungsergebnisse, die aus der Kombination von hoher Modellkenntnis und neuronalen Netzwerken herrühren, spiegeln sich auch in den Zustandsgrößen des Systems wider. Auf Grund der kaskadischen Reglerarchitektur und der Wahl der Regelgrößen ist kein Zustand des Systems Bestandteil einer internen Dynamik. Im Vergleich zur neuronalen Regelung kann etwa der Hängewinkel nicht zu unerwünschten Schwingungen angeregt werden, was in Abbildung 7.10 erkennbar ist. Auch ist der störende Einfluss der Turbulenz, der ein leicht rauschartiges Verhalten erzeugt, an diesem Verlauf gut nachzuvollziehen. Die beiden begrenzten Sprünge bei etwa 200 und 225 Sekunden markieren Phasen in der Flugbahn, in denen beim Abfangen und Steigen in Kurvenlage erhöhte Kurvenkoordination erforderlich ist. Die Sprünge werden dabei durch den linear kommandierten Höhenverlauf an den Spline-Übergängen zusätzlich begünstigt. Zusammenfassend ist festzuhalten, dass diese hochdynamische Flugbahn auch unter Störeinfluss mit guter Bahnfolgenauigkeit durchflogen werden kann, was die Robustheit des Regelungsansatzes unterstreicht. Ähnlich gutes Verhalten lässt sich auch bei der Bahnfolge der Beispielmision beobachten, was in Abbildung A.1 im Anhang dargestellt ist. Die Abweichung liegt nahezu durchgehend unter zwei Metern, wobei der zuvor gegen Ende der Mission beobachtbare Einfluss des Seitenwindes ebenso für die dynamische Inversion in abgeschwächter Form zu erkennen ist.



Abbildung 7.10: Darstellung des Hängewinkels bei erhöhtem Wind mit $u_{Wg} = 5$ m/s.

7.2 Einfluss der Lernverfahren auf die dynamische Inversion

Die bisherigen Ergebnisse der Regelung mit dynamischer Inversion sind davon ausgegangen, dass der Windeinfluss exakt gemessen werden kann, was zweifelsfrei eine stark vereinfachende Annahme ist. In [121] ist dargelegt, dass eine Bestimmung der Windvektorkomponenten auch an Bord kleiner unbemannter Systeme generell möglich ist und diese Information folglich auch dem Regelungssystem zur Verfügung gestellt werden kann. Gleichwohl treten, wie in Kapitel 4 für andere Sensoren beschrieben, unerwünschte Effekte wie zum Beispiel Rauschen bei der Messung auf. Um dieser Tatsache gerecht zu werden, sind für die Windmessung Sensorfehlermodelle implementiert worden, die von den Erkenntnissen aus Windkanal- und Flugversuchen abgeleitet sind [121]. Demnach ist davon auszugehen, dass durch die Messung zusätzliche Fehler in der Bestimmung der Windkomponenten von bis zu 0,5 m/s auftreten, was auch an dem exemplarischen Verlauf von simulierten Roh- und Messdaten aus Abbildung 7.11 für die Windkomponente u_W deutlich wird. Die Aufzeichnung dieser Daten erfolgt in der Frequenz des Regelungssystems mit 100 Hz und unterstreicht erneut den zufälligen Charakter der Turbulenz, die nun mit zusätzlichen sensorbedingten Störeinflüssen überlagert ist. Aus diesen Störungen wird deutlich, dass im Hinblick auf eine Flugversuchserprobung und die Robustheitsanalyse des Regelkreises keine Vernachlässigung von Wind und Turbulenz erfolgen darf.

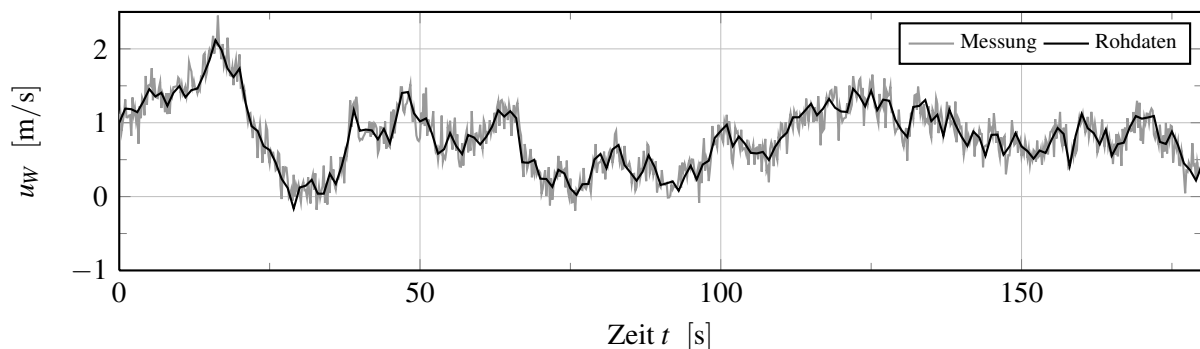


Abbildung 7.11: Darstellung des tatsächlichen und gemessenen Windes.

Im Folgenden werden die Einflüsse der beschriebenen Störungen auf die Regelung des *Carolo P360* analysiert. Dabei werden sowohl die Regelgrößen als auch die dafür maßgeblichen Stellgrößen für die Längs- und die Seitenbewegung betrachtet, um die Wirkungskette der Kaskaden zu verdeutlichen. Zu diesem Zweck werden die ersten 180 Sekunden der Flugbahn gemäß Abbildung 7.2 gewählt und zur besseren Vergleichbarkeit mit dem *P200* mit reduzierter Reisegeschwindigkeit durchflogen. Die Verläufe der einzelnen Größen sind in den Abbildungen 7.12 sowie 7.13 für die Längs- und Seitenbewegung dargestellt. Dabei wird zunächst idealisiert davon ausgegangen, dass das Systemmodell mit dem realen Flugzeug übereinstimmt. Das Netzwerktraining erfolgt mit den Gleitzustandsverfahren bei $\lambda = 50$.

Für die Längsbewegung ist die Wirkungskette von der kommandierten respektive tatsächlichen Flughöhe bis zu den dazu führenden Höhenruderausschlägen graphisch dargestellt. Zunächst

fallen zwei Aspekte besonders auf: Zum einen erfolgt der Flug mit einer sehr geringen Höhenabweichung und zum anderen ist der Einfluss der Windstörungen vor allem an den ver-rauschten Verläufen des Höhenruderausschlages η , der Nickrate q und des Bahnstellwinkels α_K deutlich. Die leichte Störung des Höhenverlaufes und der damit verknüpften Regelgrößen bis hin zur Nickrate q ist maßgeblich der Tatsache geschuldet, dass die Trimmrechnung der Simulation bei Windstille erfolgt und das Regelungssystem eine kurze Zeitspanne benötigt, um die dadurch auftretenden Ungenauigkeiten des Trimmzustandes auszuregeln.

Am Rauschverhalten der kommandierten Größen, besonders von η , q und α_K , wird der starke Einfluss der Turbulenz deutlich, wobei besonders in den inneren schnelleren Schleifen die Regelgrößen diesen Kommandos meist gut folgen können. Am Verlauf des Höhenruderausschlages ist ebenfalls der Einfluss der verzögernd wirkenden Aktuatordynamik zu erkennen. Die Regelabweichungen sind hierbei meist gering, wobei vor allem bei und nach Spline-Übergängen und damit verbundenen Änderungen der kommandierten Höhe stärkere Abweichungen auftreten. Ähnliches Verhalten trifft auch für die Nickrate q und den Bahnstellwinkel α_K zu, wobei letzterer der Sollgröße meist direkt folgen kann. Bei der in ihrer Dynamik langsameren Bahnregelungsschleife fallen qualitativ größere Abweichungen auf, die aber durchgehend beschränkt sind und gleichfalls keinen nachhaltig negativen Einfluss auf die Höhenhaltung besitzen.

In der Seitenbewegung sind der Verlauf der Regelgrößen von der Bahnabweichung d bis zur Rollrate p sowie derjenige des Querrudersignals dargestellt. Zusätzlich werden zur Verdeutlichung des Soll-Ist-Vergleiches für den Kurvenflug einzelne Bereiche von Verläufen vergrößert. An allen Regelgrößen ist zunächst zu erkennen, dass die Ist-Werte den Kommandos sehr genau folgen können; lediglich beim Querruderausschlag ξ ist der verzögernde Einfluss der Aktuatordynamik sichtbar, wobei die Abweichungen einen Grad selten überschreiten. Die laterale Bahnablage fällt in Kurven etwas höher aus als beim *P200*, was der trägeren Dynamik des *P360* bei gleichem Kurvenradius geschuldet ist. Anhand der vergrößerten Detailaufnahmen wird auch deutlich, dass Bahnazimut χ , Bahnhängewinkel μ_K und auch die Rollrate p in Kurven ein sehr genaues Folgeverhalten aufweisen. Besonders beim Geradeausflug wird am Querruderverlauf andererseits deutlich, wie dem störenden Einfluss von Wind und Turbulenz durch stetige schwache Ausschläge entgegengewirkt wird, welche zu einem schwachen Rauschverhalten in der Rollrate führen.

Insgesamt zeigt sich, dass der Störeinfluss in der Seitenbewegung sehr gut ausgeregelt werden kann und das Flugzeug ein robustes Regelverhalten besitzt. Betrachtet man ergänzend den Verlauf des Hängewinkels aus Abbildung A.2, so lässt sich erkennen, dass $\Phi \approx \mu_K$ gilt und dass zum anderen der *P360* augenscheinlich wegen seiner größeren Masse unempfindlicher gegenüber Windstörungen ist als der *P200*. Zusammengefasst kann festgehalten werden, dass das Regelungssystem bei guter Modellkenntnis eine präzise wie robuste Bahnführung auch in hochdynamischen Flugabschnitten erlaubt.

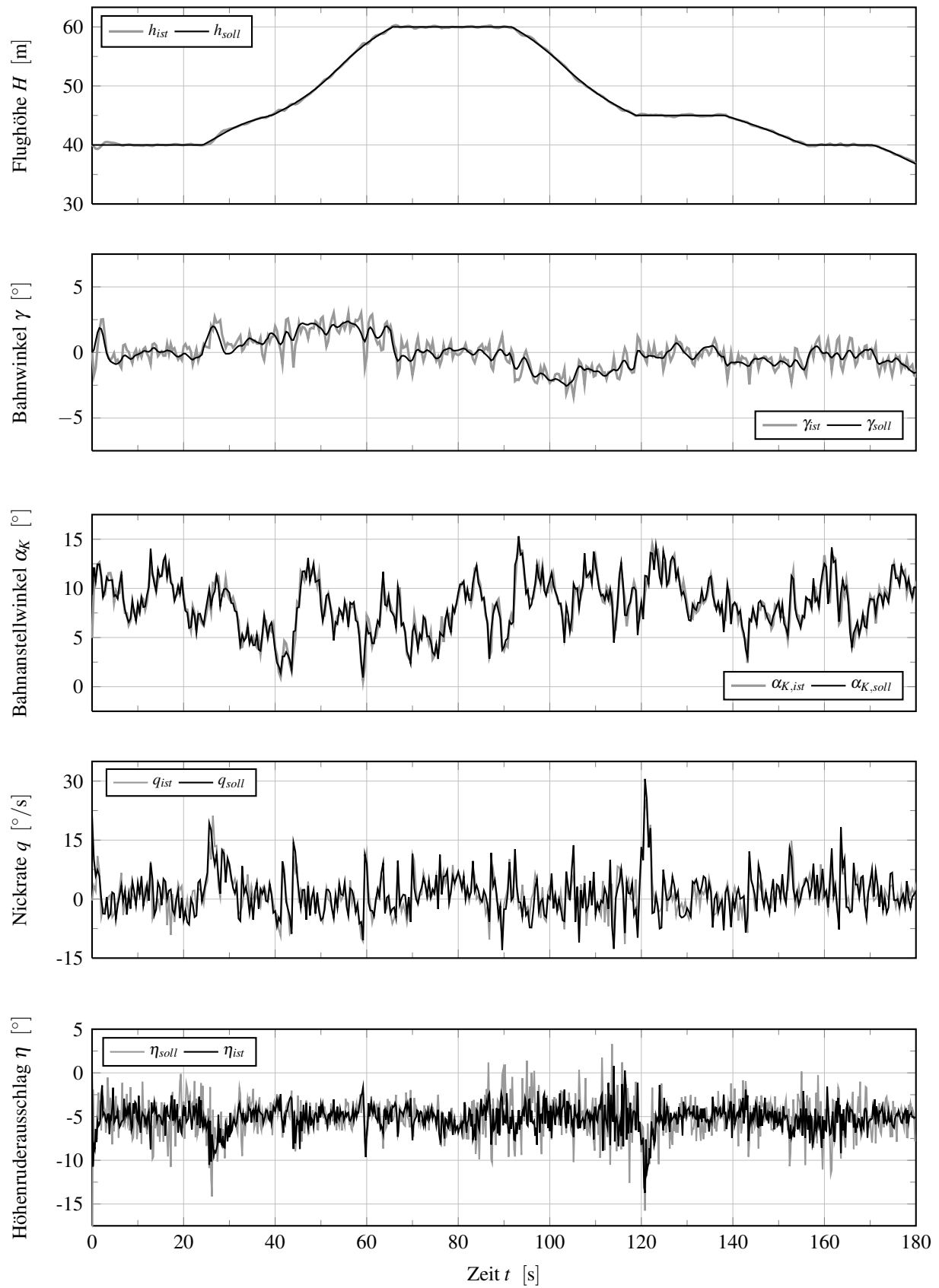


Abbildung 7.12: Größen der Reglerkaskade der Längsbewegung unter Windeinfluss.

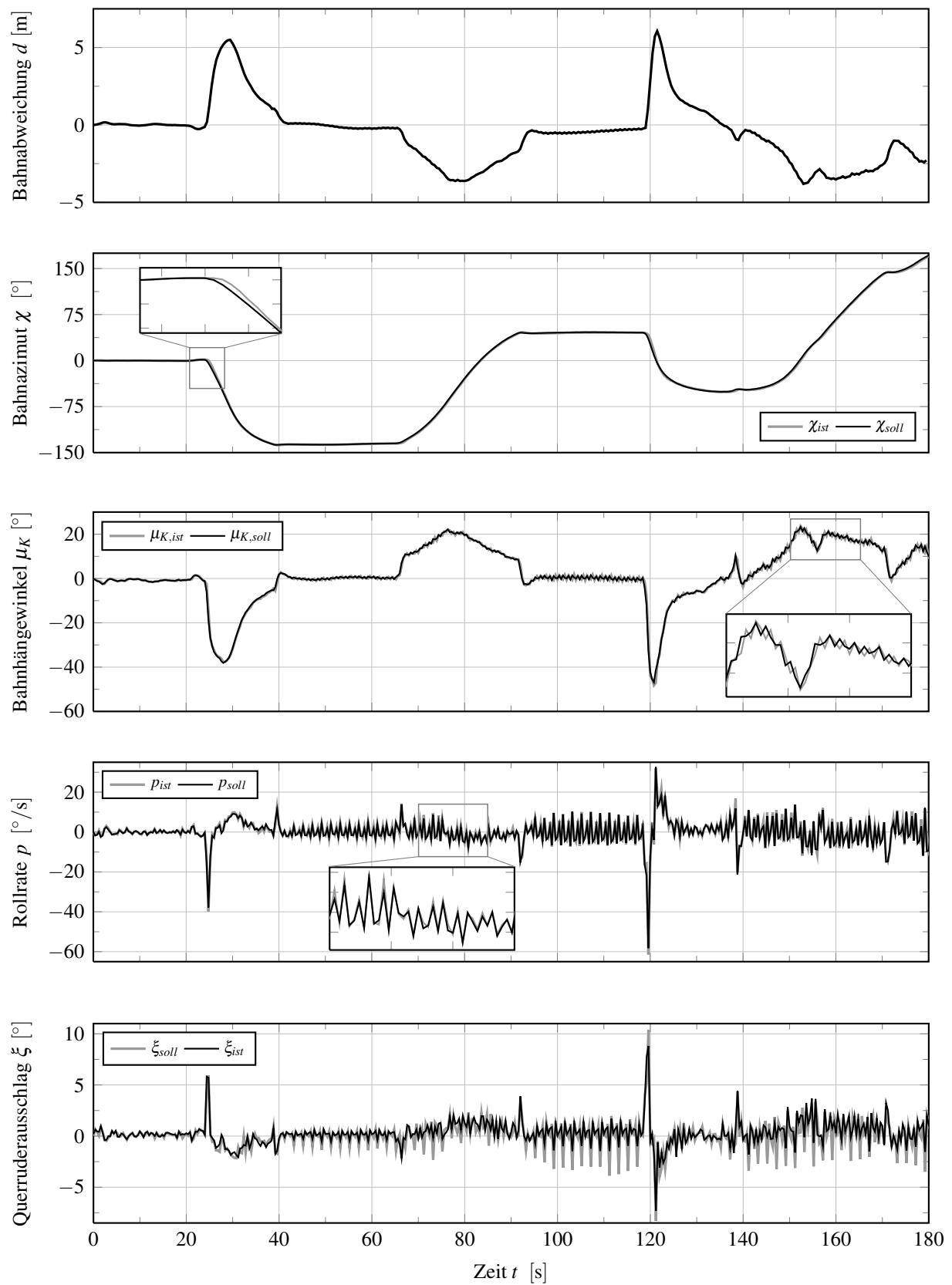


Abbildung 7.13: Größen der Reglerkaskade der Seitenbewegung unter Windeinfluss.

Da exakte Modellkenntnis in der Realität selten gegeben ist, sind im Anhang in den Abbildungen A.3 bis A.6 zusätzlich die Flugzeugreaktionen für ungenaue Modellparameter gegeben. Diese Verläufe beinhalten folglich nicht nur wie zuvor Messrauschen der Sensorik und Wind Einfluss, sondern ebenso geänderte dynamische Eigenschaften des implementierten Modells. Dazu werden die aerodynamischen Beiwerte und Trägheitsmomente pauschal um 10 % erhöht und verringert. Dieser Veränderung liegt keine direkte physikalische Motivation zu Grunde, sondern vielmehr die Demonstration des Einflusses von begrenzten Modellunsicherheiten auf den Regelkreis. Aus den Abbildungen ist ersichtlich, dass sich die Verläufe über weite Strecken nicht besonders voneinander und von denen aus den Abbildungen 7.12 und 7.13 unterscheiden, was die hohe Adaptionsfähigkeit an die eingebrachten Unsicherheiten unterstreicht. Gewisse Unterschiede sind bei dem Modell mit reduzierten Parametern in Phasen höherer Stellaktivität zu erkennen. Dieses tendiert dazu, sprunghafte Signale zu kommandieren, denen das Flugzeug nicht schnell genug folgen kann, sodass es in der Längsbewegung in diesen Phasen zu unwesentlichen, aber erkennbaren Bahnabweichungen kommt.

Neben der robusten Ausregelung von Modellunsicherheiten und atmosphärischen Störungen ist es eine wichtige Fragestellung, wie der Regelkreis auf spontan sprunghafte Inversionsfehler reagiert, wie sie zum Beispiel bei Beschädigungen des Flugzeuges auftreten. Dieses Antwortverhalten ist sowohl für bemannte als auch unbemannte Systeme von großer Bedeutung, da in solchen Extremsituationen das menschliche Reaktionsvermögen unter Umständen nicht ausreichend ist, um katastrophale Folgen zu vermeiden. Zur näheren Analyse derartiger Effekte werden in Abbildung 7.14 die Antwort des adaptiven Elementes $v_{ad,r}$, also die des neuronalen Netzwerkes v_{ad} und des robustifizierenden Terms v_r , auf einen induzierten Sprung des Inversionsfehlers in der Nickbeschleunigung von 5 rad/s^2 betrachtet. Das Training erfolgt dabei mit dem Gleitzustandsverfahren bei $\lambda = 50$. Zusätzlich wird der Einfluss dieser Antwort auf den Linearregler v_{LR} und das gesamte Reglersignal v_ω charakterisiert.

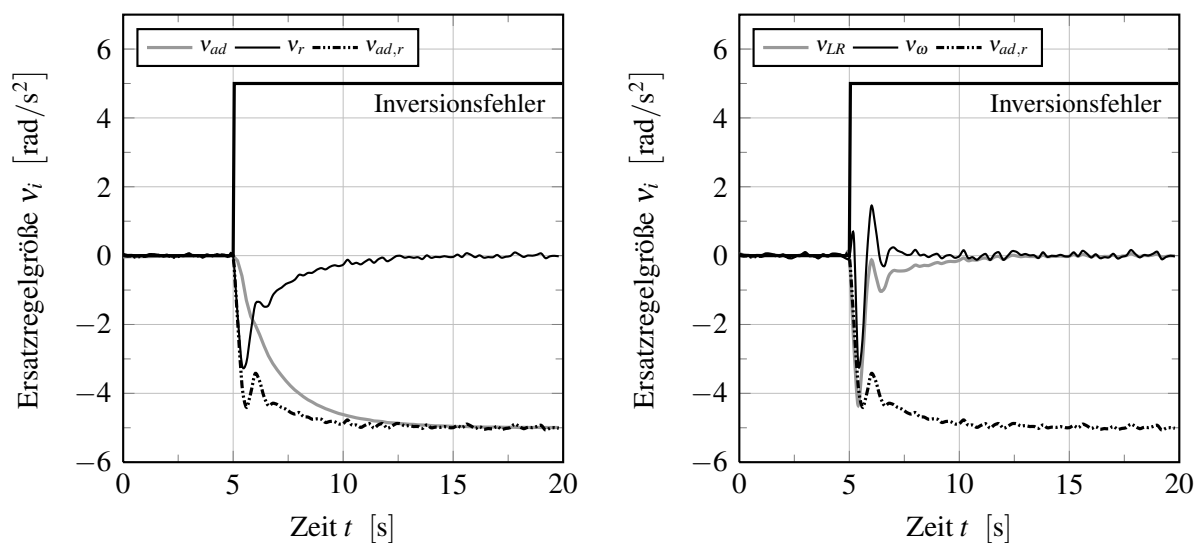


Abbildung 7.14: Sprung des Inversionsfehlers in der Längsbewegung bei idealem Flugzeug.

Das Flugzeug und seine Sensorik werden zunächst als ideal angenommen und die dynamische Beschränkung der Ersatzregelgrößen \vec{v}_h ist deaktiviert, um das Netzwerkverhalten unbeeinflusst darzustellen. Die Zeitreihe beginnt im ausgetrimmten Flugzustand und der Sprung im Inversionsfehler erfolgt bei fünf Sekunden. Es wird sich dabei im weiteren Verlauf zeigen, dass die Größenordnung des betrachteten Fehlers durchaus in einem destabilisierten Flugregelkreis auftreten kann. Im linken Teil der Abbildung ist die Reaktion des Netzwerkes und des robustifizierenden Terms sowie beider Summe dargestellt. Dabei wird deutlich, dass sich das neuronale Netzwerk in seiner Ausgabe und damit auch in seinem Fehler entsprechend der Definition der Schaltfunktion S wie ein System erster Ordnung verhält. Ebenfalls kann man den proportionalen Anteil von v_r erkennen: Ist der gefilterte Fehler groß, steigt v_r , und ist das Netzwerk an die Fehlersituation angepasst, geht v_r gegen Null. Dadurch wird die Reaktion des adaptiven Elementes auf sprunghafte Anregungen deutlich verbessert. Somit kann der Fehler binnen einer Sekunde zum größten Teil und nach fünf Sekunden gänzlich approximiert und ausgeglichen werden. Diese Eigenschaft schlägt sich auch in der Aktivität des Linearreglers v_{LR} und in dem Gesamtsignal v_ω nieder, welche im rechten Teil der Abbildung dargestellt sind. Nach einer sprunghaften Anregung kehren sowohl v_{LR} als auch v_ω in ihre Ausgangslage zurück, da der Fehler in vollem Umfang vom neuronalen Netzwerk abgefangen wird.

Dass dieses Verhalten nicht auf die Längsbewegung des Flugzeuges beschränkt ist, sondern für alle rotatorischen Freiheitsgrade gilt, wird anhand von Abbildung 7.15 deutlich. Hier wird das gleiche Fehlerszenario auf die Rollrate p angewendet. Dabei ist bei beiden Verläufen sehr ähnliches, bei der Ausgabe des adaptiven Elementes nahezu identisches Verhalten zu beobachten. Einzig das Signal des Linearreglers weist eine etwas langsamere Charakteristik auf, was auf die unterschiedliche Dynamik der Rollbewegung zurückzuführen ist. Dies unterstreicht auch, dass die Wahl eines Netzes für jede Komponente des Inversionsfehlers zweckmäßig ist.

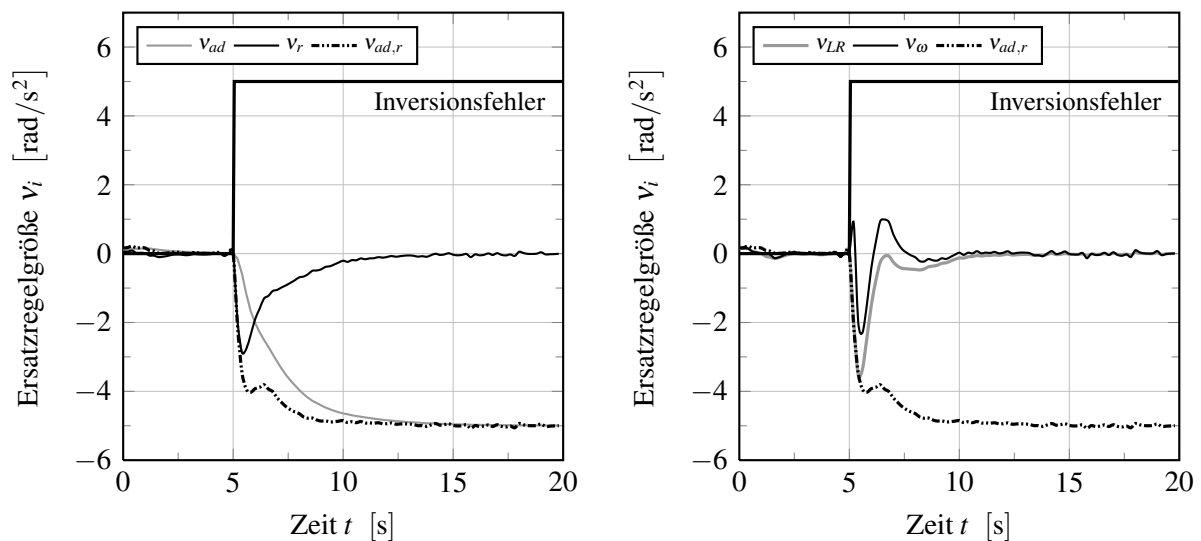


Abbildung 7.15: Sprung des Inversionsfehlers in der Seitenbewegung bei idealem Flugzeug.

Diese Betrachtung soll nun schrittweise um Modellunsicherheiten, Sensorverhalten und atmosphärische Einflüsse erweitert werden. Dazu wird die gleiche Fehlersituation der Längsbewegung und die dazugehörige Antwort des adaptiven Elementes zunächst für ein nicht exaktes Flugzeugmodell mit modellierter Sensorik betrachtet. Dies ist in Abbildung 7.16 auf der linken Seite dargestellt, wobei besonders zwei Aspekte von Bedeutung sind: Zum einen beginnt das neuronale Netzwerk im ausgetrimmten Flugzustand mit einer Ausgabe von etwa $-0,5 \text{ rad/s}^2$, womit die Abweichungen zwischen Modell und realer Strecke bereits ausgeglichen werden. Zum anderen weist der robustifizierende Term ein ausgeprägteres Rauschverhalten auf, welches auf den Einfluss des gefilterten Fehlers ζ zurückzuführen ist. Dieser wird aus den Zuständen der Fehlerdynamik berechnet, welche wiederum auf Basis der verrauschten Sensorsignale bestimmt werden. Dies unterstreicht den nachhaltigen Einfluss der nicht idealen Sensorik und des abweichenden Modells auf das Systemverhalten, wobei die Netzwerkausgabe gemäß der Störunterdrückung des Gleitzustandsverfahrens selbst davon weitestgehend unbeeinflusst bleibt. Prägnanter wird der störende Einfluss noch, wenn man zusätzlich atmosphärische Störungen und deren Messung einbezieht, was auf der rechten Seite von Abbildung 7.16 erkennbar ist. Während die Netzausgabe weiterhin relativ glattes Verhalten zeigt, nimmt der Rauschanteil im robustifizierenden Term zu, wobei dieser zum Zeitpunkt des Sprunges gleichwohl zu einer deutlichen Beschleunigung der Fehlerapproximation beiträgt. Neben der weiterhin stabilen Fehlerabbildung des Netzwerkes wird deutlich, dass die Auslegungsparameter von v_r bei nicht idealisierten Randbedingungen des Systems eine Kompromisslösung zwischen beschleunigtem Fehlerausgleich und steigendem Rauscheinfluss darstellen.

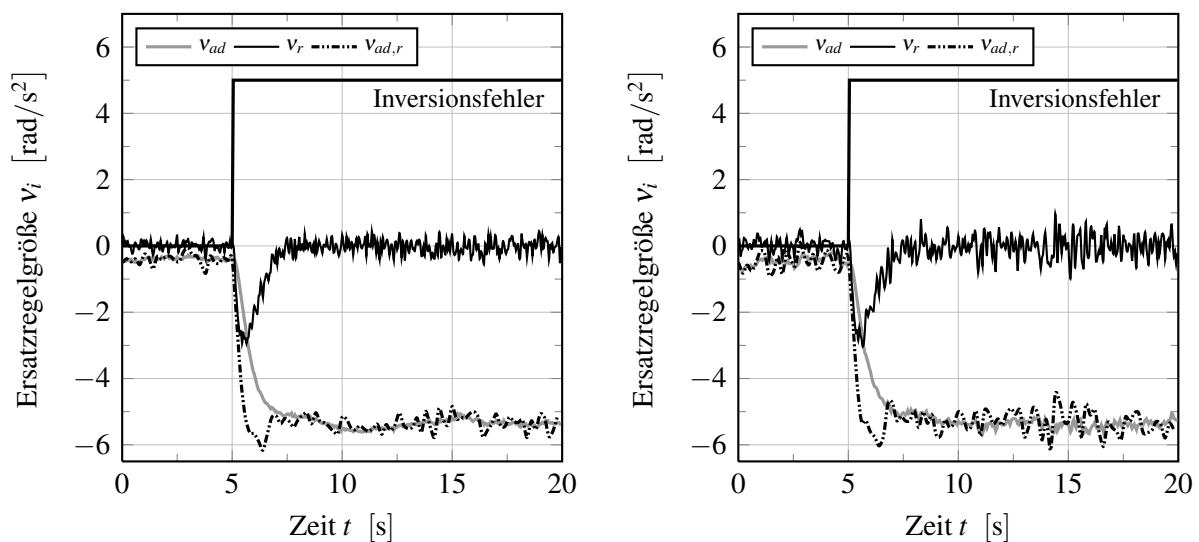


Abbildung 7.16: Sprung des Inversionsfehlers in der Längsbewegung bei Unsicherheiten.

Besonders für den Fall eines sprunghaft ansteigenden Inversionsfehlers ist die Reaktion des Gleitzustandslernverfahrens bezüglich seiner variablen Lernrate von Interesse. Ebenso relevant ist eine Betrachtung der Gewichtsnorm $\|\mathbf{w}_F\|$, da diese ein Maß für die innere Energie und damit die Stabilität des Netzwerkes ist. Hierzu sind nun in Abbildung 7.17 die Verläufe beider

Größen für besagten Sprung im Inversionsfehler dargestellt. Dabei werden sowohl Modellunsicherheiten und Sensorrauschen als auch Windstörungen gemäß Abbildung 7.16 miteinbezogen. Man sieht, dass sich die Lernrate stetig in höherem Maße ändert, solange der Netzwerkfehler gering ist. Beim Auftreten des Inversionsfehlers sinkt die Lernrate sofort drastisch um etwa eine Größenordnung ab, was in etwa auch der Änderung des Netzwerkfehlers entspricht. Dies ist schlüssig, da zu hohe Lernraten kombiniert mit einem großen Fehler zu erheblichen Gewichtsverschiebungen führen und damit destabilisierend wirken könnten. Ein schnelles Erreichen des Gleitzustandes würde somit erschwert. Auch ist es bezeichnend, dass die Verläufe von Lernrate und Norm sehr genau zu der Netzwerkausgabe aus Abbildung 7.16 passen. Die Norm erreicht ihr asymptotisch stabiles Maximum nach 2 bis 2,5 Sekunden, genau dann, wenn das Netzwerk den Inversionsfehler nahezu gänzlich approximiert hat und der Ausgabefehler und mit ihm die Gewichtsänderungen minimal sind. In dieser Zeitspanne ist auch die Lernrate minimal, da der Einfluss des höheren Netzwerkfehlers zum Erreichen des Gleitzustandes hinreichende Gewichtsänderungen bewirkt. Erst nach Ausregelung des Inversionsfehlers steigt die Lernrate wieder deutlich an, um den Verbleib des Netzwerkes im Gleitzustand zu gewährleisten. Daran wird deutlich, dass die in Kapitel 3.3 getroffenen Annahmen zu Stabilität und variabler Lernrate auch in einem geschlossenen Regelkreis Gültigkeit besitzen.

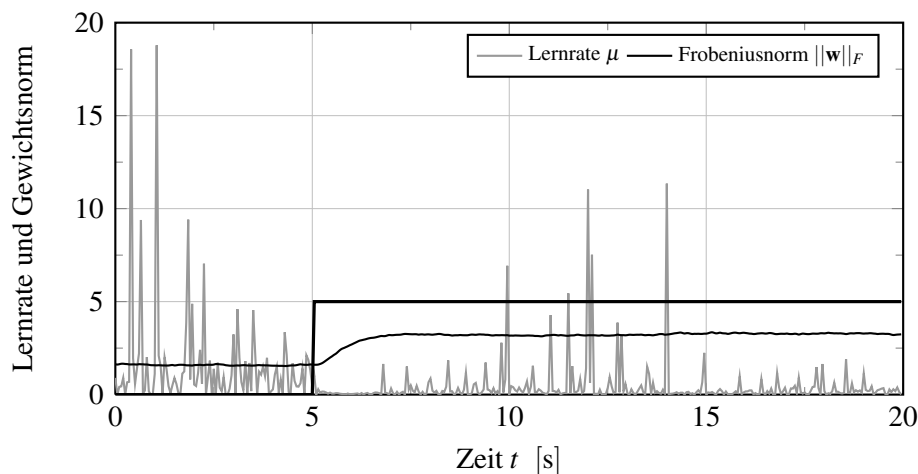


Abbildung 7.17: Sprungantwort von Lernrate und Gewichtsnorm.

Ebenfalls in Kapitel 3.3 ist die Notwendigkeit einer konservativen Auslegung der festen Lernrate des Rückpropagationsalgorithmus' dargelegt worden. Diese ist erforderlich, da ein neuronales Netz mit zu hoher Lernrate in ungünstigen Konstellationen von eigenem Zustand, Eingangssignalen und kommandierten Zielwerten unkontrolliert schwingende Ausgaben erzeugen kann. Dies gilt in besonderem Maße auch in einem mit Rückkopplungen versehenen Regelkreis. Um besagten Effekt zu illustrieren, wird das Flugzeug durch einen ähnlichen sprunghaften Inversionsfehler in der Längsbewegung beeinträchtigt, wobei dieser mit 3 rad/s^2 moderater gewählt ist. Dabei wird dieselbe Bahn wie in den Abbildungen 7.12 und 7.13 durchflogen. Die Beschränkung der Ersatzregelgröße durch das PCH bleibt zunächst deaktiviert, was trotz seiner prinzipiell abschirmenden Wirkung aus zwei Gründen erfolgt: Einerseits sollen

die potentiell möglichen Schwingungen der Netzausgabe sichtbar werden und andererseits hat ein Inversionsfehler dieses Ausmaßes auch direkten negativen Einfluss auf das PCH. Sein Signal v_h , das die Signalvorgabe der Referenzmodelle auf ein Maß begrenzen soll, dem das System folgen kann, wird aus Modellgrößen bestimmt. Je größer nun der Unterschied zwischen Modell und realer Strecke, desto ungenauer kann das begrenzende Signal \vec{v}_h bestimmt werden. Dieser Effekt wird weiter gesteigert, wenn die für die Berechnung nötigen Ruderausschläge nur ungenau mess- oder schätzbar sind. In Extremsituationen können so sich gegenseitig anfachende Schwingungen entstehen, was vor allem durch eine konservative Wahl der festen Lernrate μ verhindert werden kann.

Die resultierenden Verläufe der gefilterten Fehlergröße ζ für das mit Unsicherheiten und Störungen beaufschlagte Fluggerät sind in Abbildung 7.18 für das Gleitzustandsverfahren (GZR) und den Rückpropagationsalgorithmus (RP) gegeben. Die Wahl von $\lambda = 50$ für die Schaltfunktion ist moderat; die Lernrate des RP-Verfahrens ist mit $\mu = 0,2$ empirisch ermittelt, wobei alle höheren Werte einen qualitativ ähnlichen Verlauf erzeugen. Alle Parameter, die von beiden Verfahren genutzt werden, wie etwa e-Modifikation und robustifizierender Term sind hierbei identisch. Es ist zu erkennen, dass der Sprung bei Sekunde Fünfzehn erfolgt und beide Netzwerke zunächst mit einer prinzipiell ähnlich guten Fehlerkompensation reagieren.

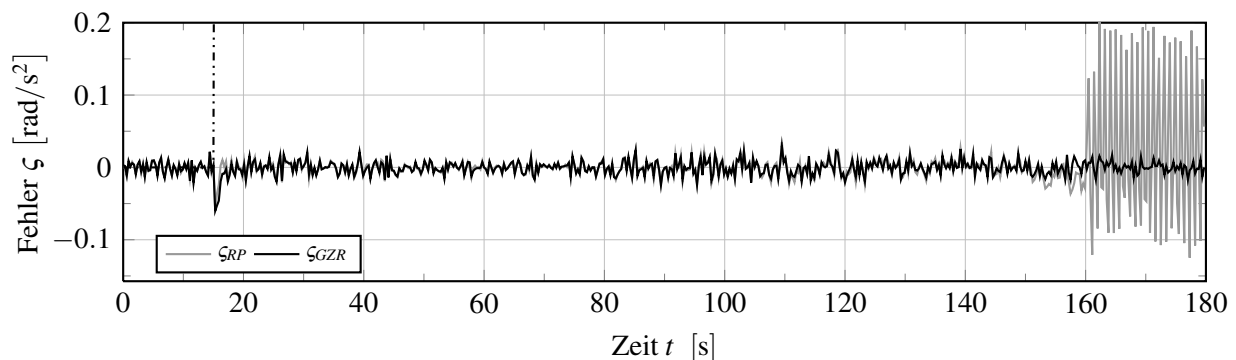


Abbildung 7.18: Destabilisierung des Netzwerkes bei zu hoher fester Lernrate.

An dieser Stelle sei ergänzend zu Kapitel 3.3 angemerkt, dass es häufig möglich ist, die Lernrate des Rückpropagationsalgorithmus' für ein bestimmtes Lernproblem empirisch so anzupassen, dass es genauso gute oder zum Teil gar leicht bessere Ergebnisse als das Gleitzustandsverfahren liefert. Die Unwägbarkeit besteht darin, dass diese für den einen Fall adäquat gewählte Lernrate für ein anderes Lernproblem zu Instabilitäten führen kann. Betrachtet man ein Flugzeug, seine Enveloppe und die verschiedenen nichtlinearen Flugzustände, die durch auftretende Extremsituationen noch einmal massiv verändert werden können, wird klar, dass eine feste Lernrate schwerlich für alle Eventualitäten optimal ermittelt werden kann. Sie kann für den einen Flugzustand ideal und für einen weiteren deutlich zu hoch sein. Genau dieser Fall tritt trotz eines moderaten Fehlers in Abbildung 7.18 ab etwa 150 Sekunden ein, wobei kein

besonderes Manöver in der Längsbewegung vollzogen wird. Der Ausgabefehler ζ_{RP} und mit ihm die Ausgabe v_{ad} beginnen erst schwach und dann unkontrolliert zu schwingen, wobei die Amplitude der Schwingung sogar ausreicht, um das Flugzeug gänzlich zu destabilisieren. In identischer Situation weist das Gleitzustandsverfahren dieses Verhalten nicht auf. Zu begründen ist dies mit dem gleichen Effekt wie aus Abbildung 7.17 und der dazugehörigen Erklärung: In Abhängigkeit des Netzwerkfehlers wird die Lernrate so angepasst, dass der Gleitzustand erreichbar bleibt und die Zustände des Netzwerkes nicht unkontrolliert über die Schaltlinie hinausstreben. Ein weiterer großer Vorteil des Gleitzustandsverfahrens besteht folglich darin, dass es die Lernrate des Netzwerkes selbsttätig an den aktuellen Flugzustand anpasst und damit eine aufwändige und dennoch konservative Auslegung einer festen Lernrate entfällt.

Nachdem das unterschiedliche Verhalten der Lernverfahren anhand sprungförmiger Inversionsfehler diskutiert worden ist, sollen nun die Reaktionen auf simulierte Schadensfälle analysiert und damit die Leistungsfähigkeit der gesamten Regelungsstrategie bewertet werden. Da eine exakte aerodynamische Modellierung von Beschädigungen am Flugzeug sehr komplex ist, sollen stellvertretend hierfür stattdessen einzelne Derivative, die starken Einfluss auf die Systemdynamik haben, verändert werden. Um ein anspruchsvollen Fehlerszenario zu gewährleisten, werden sowohl Größen der Längs- als auch Seitenbewegung verändert und das Flugzeug einer moderaten Windstörung ausgesetzt. In der Längsbewegung wird das Flugzeug statisch destabilisiert, indem die eigentlich negative Steigung von $C_{m\alpha}$ ins Positive gedreht und die Nickdämpfung C_{mq} um 50 % reduziert werden. In der Seitenbewegung wird gleichzeitig ein starker Eingriff in die statische Stabilität vorgenommen, indem die Steigung der Windfahnenstabilität $C_{n\beta}$ negativ angenommen wird. Zusätzlich werden wie in der Längsbewegung die Rolldämpfung C_{lp} und die Gierdämpfung C_{nr} um 50 % reduziert. Dieses simulierte Fehlerszenario ist in seinen Auswirkungen tatsächlich so gravierend, dass der nicht durch neuronale Netzwerke gestützte Regelkreis sofort instabil wird und das Flugzeug nach einem Strömungsabriss zum Absturz kommt.

Die Simulation beginnt im Trimmzustand des unbeschädigten Flugzeuges und mit sofort wirksamen Derivativänderungen, was bedeutet, dass die aus der Beschädigung resultierenden Inversionsfehler sprunghaft ansteigen. Daraus folgt, dass das sich im Reiseflug befindliche Flugzeug durch einen schweren Schadensfall abrupt destabilisiert wird. Dabei durchfliegt der P360 die bereits diskutierte Bahn aus Abbildung 7.2 mit normaler Reisegeschwindigkeit von 23 m/s und gestützt durch Netzwerke trainiert mit dem Gleitzustandsverfahren. Außerdem werden leicht erhöhte atmosphärische Störungen als zuvor angenommen und die von der Turbulenz überlagerten Windkomponenten mit $u_W = 3$ m/s, $v_W = 2$ m/s sowie $w_W = 1$ m/s gewählt. Die Verläufe der Stell- und Regelgrößen sind analog zu den vorherigen Testfällen in den Abbildungen 7.19 und 7.20 für die Längs- und Seitenbewegung dargestellt.

Betrachtet man den Höhenverlauf aus Abbildung 7.19, so fallen neben einer leicht reduzierten Bahnfolgenauigkeit zwei Schwingungen in kritischen Flugzuständen auf. Die erste ist die

direkte Folge der durch die Beschädigung auftretende statische Destabilisierung, deren Ausregelung sofort einsetzt und im weiteren Verlauf somit größere Höhenabweichungen verhindert. Die zweite stärkere Schwingung erfolgt an dem bereits zuvor diskutierten anspruchsvollen Abfangmanöver im Kurvenflug. Dass die signifikante Beeinflussung der Längsbewegung zu kritischen Flugzuständen führt, wird auch am Verlauf der Winkel γ und α_K deutlich. Beide zeigen das zu erwartende sofortige Aufnicken des Flugzeuges nach Auftreten des Schadens, dem der Regler direkt entgegenwirkt. Dabei wird ein kritisches Aufnicken bis zum Strömungsabriss nur durch das schnelle zusätzliche Entgegenwirken der neuronalen Netze verhindert, da der ungestützte Regelkreis genau an dieser Stelle instabil wird. Wie zu erwarten zeigen beide Verläufe unruhigeres Verhalten mit größeren Soll-Ist-Abweichungen als diejenigen aus Abbildung 7.12. Besonders fallen die starken Ausschläge bei besagter Schwingung im Höhenverlauf auf, da der Bahnstellwinkel auf Grund des ungünstigen Flugzustandes kombiniert mit der Beschädigung kurzzeitig deutlich über die Beschränkung hinausläuft. Die Nickrate q weist höhere kommandierte Werte als im Nominalfall auf, zeigt aber prinzipiell einen ähnlichen qualitativen Verlauf. Dies gilt von einer Besonderheit abgesehen auch für den Höhenruderausschlag η . Es ist gut zu erkennen, dass der Ausschlag im Vergleich durchgehend um etwa 5° verschoben ist, was mit einer erlernten, netzwerkbedingten Anpassung der Höhenrudertrimmung an die Beschädigung des Flugzeuges zu erklären ist.

Die Auswirkungen in der Seitenbewegung sind weniger drastisch, was in Abbildung 7.20 deutlich wird. Die laterale Bahnablage ist erwartungsgemäß zwar höher als im Nominalfall, fällt trotz der fehlenden Windfahnenstabilität mit Zusatzabweichungen im Meterbereich allerdings deutlich begrenzt aus. Die negativen Effekte auch der reduzierten Dämpfungen können durch das Regelungssystem augenscheinlich gut kompensiert werden. Besonders an den Verläufen von μ_K , p und ξ ist ersichtlich, dass die veränderten Derivative zu erhöhten Kommandos in Kurvenlagen führen, was besonders beim Abfangmanöver bei etwa 110 Sekunden zu einem unerwünschten Überschwinger auf 60° Querlage führt. Besonders aber beim Geradeausflug fällt an den drei besagten Verläufen auf, dass diese trotz Windes ruhigeres Verhalten aufweisen als im Nominalfall. Dies ist zum einen damit zu erklären, dass durch die reduzierten Dämpfungen auch reduzierte, den Drehbewegungen entgegenwirkende Momente entstehen, das Flugzeug also prinzipiell agiler wird. Damit muss die Kombination aus Linearregler und adaptivem Element zur Bahnfolge in diesen Flugabschnitten geringere Sollwerte kommandieren, denen das Flugzeug seinerseits mit geringerer Verzögerung zu folgen vermag. Zum anderen wird ähnlich wie in der Längsbewegung zum Ausgleich der reduzierten statischen Stabilität eine stetige Neutrimmung des Seitenruders vorgenommen, was zu einem Flug mit erhöhtem Schiebewinkel führt und in Kombination mit den Querrudern zu dem dargestellten Verlauf führt.

Der nachhaltige Einfluss der neuronalen Netzwerke auf die Stabilisierung des Flugzeuges wird besonders bei Betrachtung der Netzausgabe deutlich. Diese ist für das Netzwerk der Nickbewegung in Abbildung 7.21 dargestellt. Zur besseren Einschätzung der Größenordnung der Signale von Linearregler und neuronalem Netzwerk in Gegenwart großer Inversionsfehler sei hierzu

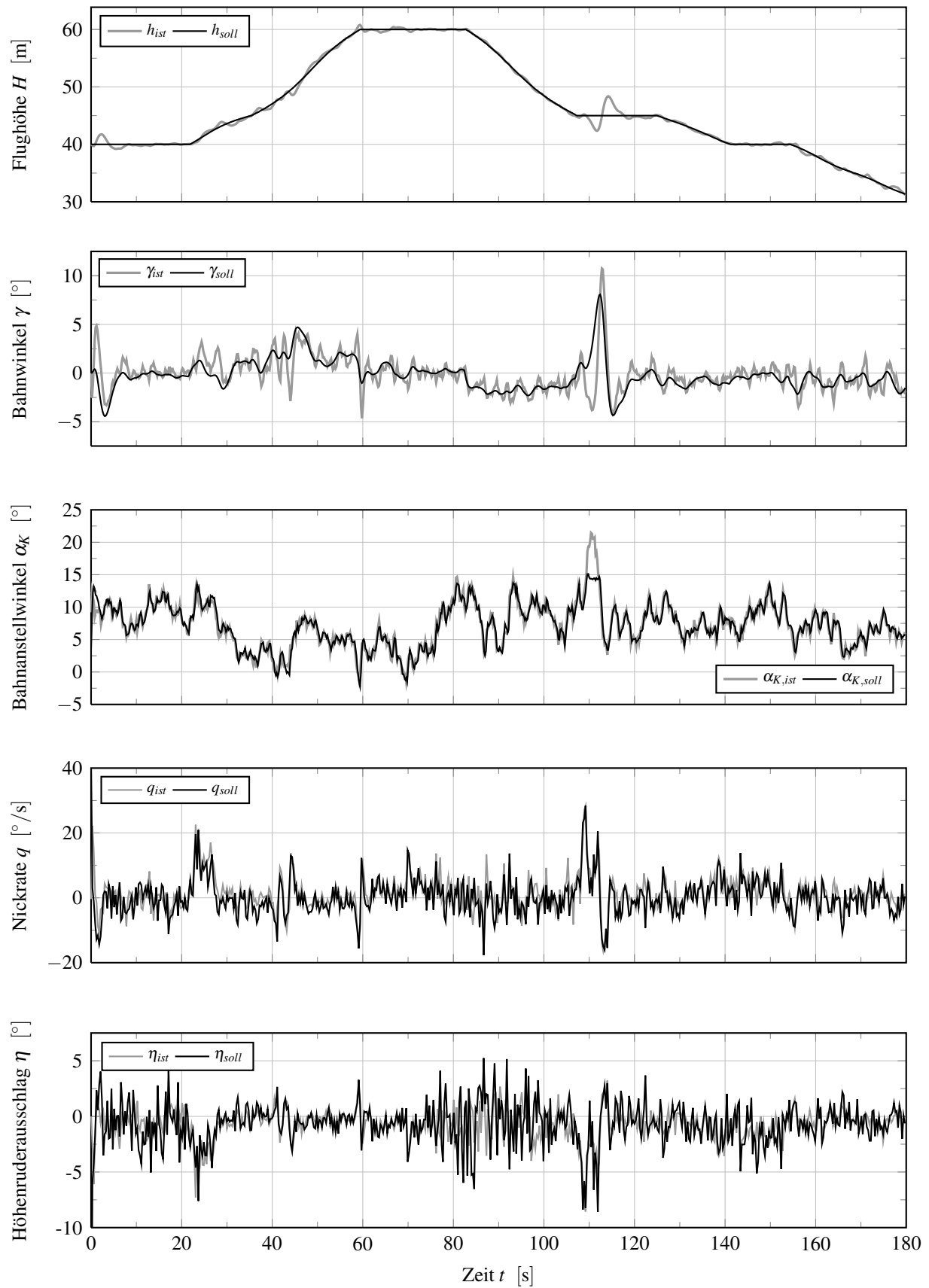


Abbildung 7.19: Größen der Reglerkaskade der Längsbewegung mit Schadensfall.

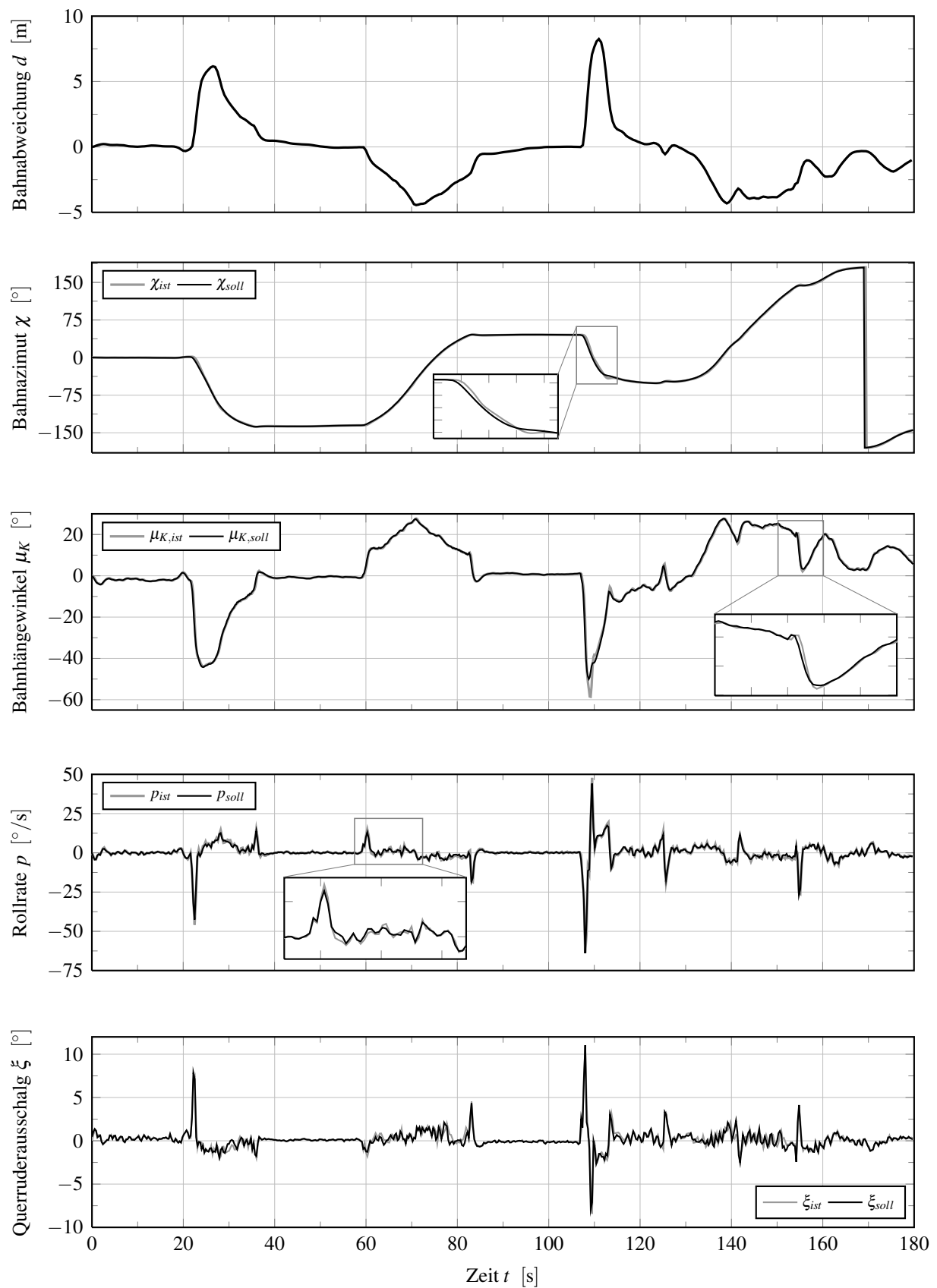


Abbildung 7.20: Größen der Reglerkaskade der Seitenbewegung mit Schadensfall.

noch einmal auf Abbildung 7.14 verwiesen. Daran ist ersichtlich, dass die Destabilisierung der Längsbewegung einen dem Sprung aus Abbildung 7.14 in seiner Größe ähnlichen Inversionsfehler produziert. Gleichmaßen wird am Verlauf des Ausgabesignals deutlich, dass das Netzwerk sofort auf den sprunghaften Anstieg des Fehlers auf etwa 4 rad/s^2 reagiert und diesen dauerhaft ausgleicht. Während sich dabei das Signal des Linearregler im Vergleich zum Nominalzustand qualitativ wenig ändert, erhält das des Netzwerkes einen dominierenden Einfluss.

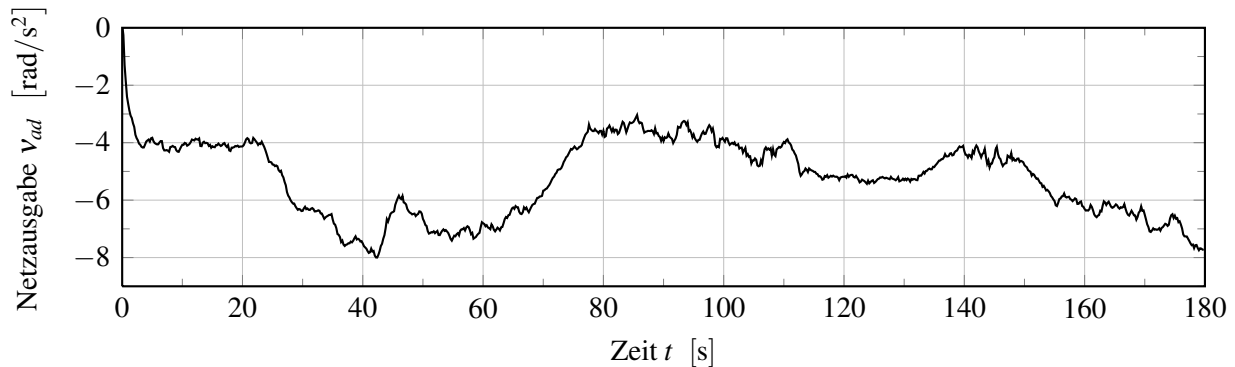


Abbildung 7.21: Ausgabe des Netzwerkes zur Stützung der Nickgeschwindigkeit.

Dass die Netzwerkeigenschaften in letzter Konsequenz auch einen erheblichen Einfluss auf die automatische Bahnfolge haben, wird deutlich, wenn man die verschiedenen Lernverfahren unter den gegebenen Fehlerbedingungen vergleicht. Hierzu sind in Abbildung 7.22 nun die Höhenabweichungen für feste Lernraten von $\mu = 0,1$ und $\mu = 0,075$ sowie für das Gleitzustandsverfahren gegeben. Die Wahl der Lernraten orientiert sich dabei an den Ergebnissen aus Abbildung 7.18 und berücksichtigt die Forderung nach ihrer stabilitätsbedingten konservativen Auslegung.

Man erkennt an der Reaktion auf den eingetretenen Schaden, welchen starken Einfluss die Konvergenzgeschwindigkeit des Lernverfahrens auf den Höhenverlauf besitzt. Während das Durchsacken und anschließende Überspringen beim Gleitzustandsverfahren mit etwa -2 und 1 Meter begrenzt ausfällt, steigen diese Werte je nach Lernrate auf das Doppelte bis Dreifache. Hieran ist leicht nachzuvollziehen, dass eine zunehmend geringere Wahl der Lernrate schnell zu einer Reglerkonfiguration führt, die den existierenden Schaden nicht mehr kompensieren kann und wegen zu starken Auf- und Abnickens zum Strömungsabriss führt. Haben sich die unterschiedlich trainierten Netzwerke erst einmal an den Inversionsfehler adaptiert, zeigen die Höhenverläufe bei begrenzter Dynamik der durchgeführten Flugmanöver qualitativ kaum Unterschiede. Einzig bei besagtem Abfangmanöver zeigt das Gleitzustandsverfahren mit maximalen Abweichungen von $-3,38$ und $2,65$ Metern erneut erkennbar bessere Ergebnisse als die Netze mit fester Lernrate; diese zeigen für $\mu = 0,1$ Abweichungen von $-3,72$ und $2,79$ sowie für $\mu = 0,075$ Höhenfehler von $-4,04$ und $3,0$ Metern. Auch wenn die Unterschiede

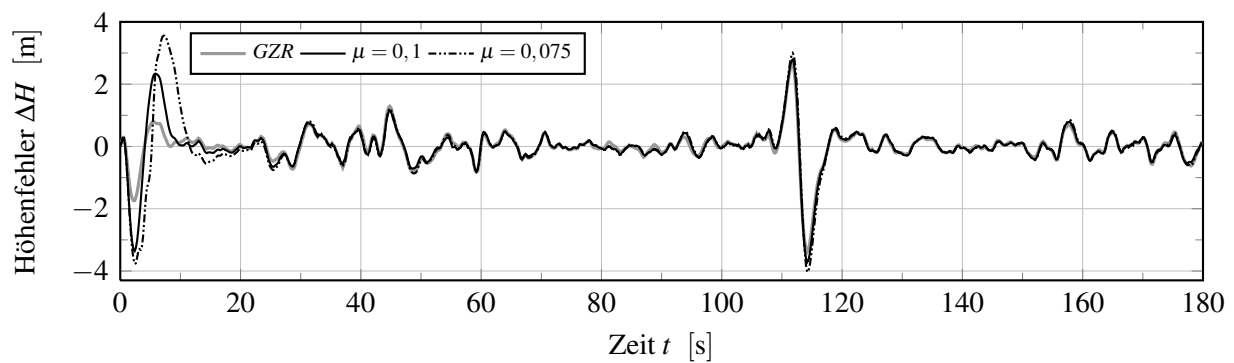


Abbildung 7.22: Vergleich der Höhenabweichung der verschiedenen Lernverfahren.

zunächst gering ansetzen, zeigt sich erneut eine Leistungseinbuße bei konservativ zu wählender Lernrate und dass weiterhin auch nach der Anpassung der Netze an bestehende Inversionsfehler das Gleitzustandsverfahren eine schnellere Reaktion auf kritische Flugzustände erlaubt. In Abbildung A.7 im Anhang ist außerdem ersichtlich, dass ähnliches Verhalten auch in der Seitenbewegung erzeugt werden kann. Da die Seitenbewegung für den betrachteten Fall insgesamt stabiler geregelt werden kann, beginnen deutliche Unterschiede im Bahnverlauf erst ab einer Lernrate von $\mu = 0,075$.

Die Ergebnisse zur automatischen Bahnführung eines stark destabilisierten Flugzeuges belegen die Leistungsfähigkeit dieses lernfähigen Regelungssystems. Dass ein derart beeinträchtigtes Flugzeug mit der vorgestellten Regelungsstrategie auch eine längere Flugmission erfolgreich durchführen kann, ist anhand der Abbildungen A.8 und A.9 zu erkennen. Hier sind die Verläufe des Höhenfehlers und der seitlichen Bahnabweichung für die wissenschaftliche Flugmission aus Abbildung 7.7 wie zuvor unter Verwendung des Gleitzustandsverfahrens bei Windstörungen und Modellunsicherheiten dargestellt. Man erkennt, dass es nach der anfänglichen Adaption an den Inversionsfehler nur zu marginalen Einbußen in der Bahnführungsgenauigkeit kommt und die Durchführung der Mission davon unbeeinflusst bleibt. Dies unterstreicht nicht zuletzt die langzeitstabile Bestimmung der Lernrate durch das Gleitzustandslernverfahren.

8 Zusammenfassung und Ausblick

In der vorliegenden Arbeit werden systematische Ansätze zur Nutzung neuronaler Netzwerke in nichtlinearen adaptiven Regelungsverfahren vorgestellt. Ausgangspunkt der Betrachtungen sind die theoretischen Grundlagen vorwärtsgerichteter neuronaler Netze, ihre Approximationseigenschaften, die Fähigkeit dynamische Prozesse abzubilden sowie ihre gängigen Lernverfahren. Dabei wird deutlich, dass ein statistischer Lernansatz, der große Gruppen von Netzwerken für ein Lernproblem nutzt, zur ingenieurstechnischen Reproduzierbarkeit und Bewertung des Trainingserfolges maßgeblich beiträgt. Ebenso werden die Bedeutung der Auswahl der Trainingsdaten und ihrer Aufbereitung sowie die Bewertung von Generalisierungsfähigkeit und Robustheit besonders unterstrichen. Diese Schritte tragen dazu bei, den Entwurfsprozess neuronaler Netzwerke stärker im Kontext klassischer Software-Verifikation und -Validation zu begreifen und so ihre Leistungsfähigkeit zu verbessern.

Für die regelungstechnische Anwendung neuronaler Netze besteht die Herausforderung darin, die Einflüsse des Lernprozesses auf einen geschlossenen Regelkreis im Sinne der Stabilität zu quantifizieren. Um die daraus üblicherweise folgende konservative Auslegung der Lernverfahren zu umgehen und um neuronale Netzwerke regelungstechnisch transparenter begreifbar zu machen, wird ein Paradigmenwechsel bei ihrer systemtechnischen Betrachtungsweise vollzogen. Ein Netz wird nicht länger als zu optimierendes System betrachtet, sondern selbst als geregeltes System aufgefasst. Dadurch ist es möglich, das Konzept der nichtlinearen strukturvariablen Regelung und ihren Spezialfall der Gleitzustandsregelung auf neuronale Lernverfahren zu übertragen. Dieser Ansatz zwingt die Zustände des geregelten Systems in den sogenannten Gleitzustand, der seinerseits global asymptotisch stabil ist. Durch die Betrachtung eines neuronalen Netzes als dynamisches System können also letztlich seine Zustandsgrößen geregelt und ihre Trajektorien in klassischen Phasendiagrammen analysiert werden. Dabei erfordert die regelungstechnische Auslegung dieses Lernverfahrens, dass die Lernraten zur garantierten Erreichbarkeit des Gleitzustandes dynamisch bestimmt werden. Dadurch verfügt der Lernansatz über eine regelungstechnisch motivierte, intrinsische Stabilitätsforderung bei gleichzeitiger höherer Robustheit gegenüber externen Störungen. Diese im Rahmen der Theorie hergeleiteten Eigenschaften werden schließlich in exemplarischen Trainingssituationen analysiert und bestätigt.

Im Anschluss werden die grundlegenden Annahmen für die nichtlineare Flugsimulation, mit der die Validierung und Bewertung der lernfähigen Regler erfolgt, dargelegt. Zur Vorbereitung

der Flugerprobung der diskutierten Regelungsansätze wird dabei auch auf die Modellierung im realen Flugzeug existierender Störeinflüsse eingegangen. Dazu wird neben Sensorfehlermodellen, die verrauschte und fehlerbehaftete Messsignale liefern, auch ein Störgrößenmodell genutzt, welches ein *Dryden*-Turbulenzspektrum beinhaltet. Somit wird das Flugzeug nicht idealisiert betrachtet und das Maß der auf den Regelkreis wirkenden Störungen kann deutlich realistischer gestaltet werden.

Der erste lernfähige Regelungsansatz, der anschließend vorgestellt wird, ist die modellfreie neuronale Regelung. Wie der Name es andeutet, ist das Besondere dieser Regelungsstrategie, dass kein mathematisches Modell des zu regelnden Systems vorhanden zu sein braucht. Alle für die Regelung nötigen dynamischen Eigenschaften können von zwei miteinander verschalteten neuronalen Netzen erlernt werden: einem Regler- und einem Beobachternetzwerk kombiniert mit einem linearen Referenzmodell zweiter Ordnung. Am Beispiel der Seitenbewegung eines unbemannten Flugsystems wird gezeigt, dass der neuronale Beobachter das dynamische Verhalten einer Regelgröße stabil und genau zu schätzen vermag. Dieser kann durch Rückpropagation des Regelfehlers so die Funktion einer invertierten Jacobi-Matrix von Regel- und Stellgrößen übernehmen, womit schließlich dem neuronalen Regler ein stabiles Trainingssignal für das Lernen im Flug bereitgestellt werden kann. Außerdem wird der zuvor eingeführte systematische Entwurfsprozess auf die Auslegung der Regler- und Beobachternetzwerke angewendet, um diesen im Vorfeld des Einsatzes ein fundiertes Basiswissen anzutrainieren. Dabei zeigt sich, dass vorwärtsgerichtete Netzwerke die komplexen, nichtlinearen dynamischen Zusammenhänge der Flugregelung auch unter Einfluss von atmosphärischen Störungen und Sensorrauschen sehr genau erlernen können. Ebenso unterstreicht die Analyse der Generalisierungsfähigkeit der verwendeten Netze, dass die zur Regelung nötigen Eigenschaften auch in robuster Weise erlernt werden. Die statistische Auswertung einer großen Gruppe trainierter Netzwerke erlaubt schließlich auch die Eingrenzung der Netzwerkgröße und die konkrete Auswahl im Regelkreis zu implementierender Netzwerke. Damit wird ein reproduzierbarer Entwurfsprozess lernfähiger, arbeitspunktunabhängiger Regelungssysteme geschaffen, der sich besonders für unbemannte Kleinflugzeuge eignet.

Mit der nichtlinearen dynamischen Inversion wird danach ein Regelungsverfahren vorgestellt, das in hohem Maße auf physikalischer Modellkenntnis basiert. Dieser Ansatz ermöglicht es, durch eine Kombination aus nichtlinearer Koordinatentransformation und Zustandsrückführung ein nichtlineares System in ein arbeitspunktunabhängiges lineares Ersatzsystem zu überführen. Neben der Definition des relativen Grades als Voraussetzung der Inversion werden die theoretischen Hintergründe von Koordinatentransformation, Zustandsrückführung und nicht zuletzt interner Dynamik hergeleitet und interpretiert. Mit der Etablierung des linearen Ersatzsystems werden der Inversionsfehler als Abweichung zwischen Streckenmodell und realem System sowie die Fehlerdynamik zur Charakterisierung des daraus resultierenden Systemverhaltens eingeführt. Zur Beeinflussung dieser Fehlerdynamik kann in Folge ein linearer Regelungsansatz verwendet werden, der zur Berücksichtigung des nicht invertierten

Aktuatorverhaltens um eine dynamische Beschränkung der Regelgröße des Ersatzsystems, das sogenannte *pseudo control hedging*, erweitert wird. Diese Konfiguration stellt die Grundlage des adaptiven Regelungssystems dar.

Um existierende Inversionsfehler, die aus ungenauer Systemmodellierung, Degradation oder Beschädigungen herrühren, auszugleichen, erfolgt anschließend die Erweiterung der dynamischen Inversion mit neuronalen Netzwerken. Dazu wird zunächst der Standardansatz mit dem Rückpropagationsalgorithmus als Lernverfahren hergeleitet und im Sinne der Stabilität der neuronal beeinflussten Fehlerdynamik um die Mechanismen der e-Modifikation und des robustifizierenden Terms erweitert. Dabei ist durch die kombinierte Nutzung einer geeigneten Ljapunow-Funktion und der hergeleiteten Lernregeln der Stabilitätsnachweis des geschlossenen Regelkreises möglich. Diese Regelungsstrategie im weiteren Verlauf mit dem Gleitzustandslernverfahren zu verbinden, stellt letztlich einen der Schwerpunkte der vorliegenden Arbeit dar.

Dadurch nämlich, dass die gefilterte Fehlergröße, welche direkt aus den Zuständen der neuronal beeinflussten Fehlerdynamik abgeleitet ist, das Lernsignal darstellt, wird diese zur Zustandsgröße des neuronalen Netzwerkes. Somit wird der gefilterte Fehler durch das Lernverfahren in den Gleitzustand gezwungen, was bedeutet, dass auch die Zustände der Fehlerdynamik und ihre Zeitableitungen in den Gleitzustand verbracht werden. Es zeigt sich, dass damit eine neue Sichtweise auf die Stabilitätsanalyse neuronal beeinflusster Inversionsregler möglich ist und dass ein lernendes Netzwerk aktiv zur Stabilität des geschlossenen Regelkreises beizutragen vermag. Als weiterer Vorteil des Gleitzustandsverfahrens wird außerdem die automatische Bestimmung der Lernrate identifiziert, da dadurch die empirische oder auf starken Idealisierungen basierte Bestimmung eines festen Wertes derselben entfällt. Bevor schließlich die hergeleiteten Regelungsansätze durch Simulationen validiert und bewertet werden, erfolgt die Beschreibung und Auslegung des Flugregelkreises als Kaskade dreier Schleifen mit relativem Grad Eins sowie die Diskussion der Reglerparameter.

Die simulative Bewertung der Regelungsstrategien geschieht durchgehend unter der Verwendung der vorgestellten Sensormodelle und unter Einfluss von Wind und Turbulenz, womit möglichst realistische Störungen erzeugt werden sollen. Dabei zeigt sich, dass die modellfreie neuronale Regelung entsprechend der theoretischen Annahmen in der Lage ist, das nichtlineare, verkoppelte Systemverhalten eines unbemannten Fluggerätes auch unter Störeinfluss stabil und genau zu regeln. Dies wird auch bei der automatischen Bahnfolge für eine exemplarische Flugmission mit längerer Flugzeit deutlich. Trotz der Komplexität des Lernproblems, welches explizit nicht nur Standardmanöver oder ausgewählte Referenzsignale wie etwa Sprunganregungen beinhaltet, erlaubt diese Reglerarchitektur, das antrainierte Basiswissen im Flug über eine breite Enveloppe zu erweitern. Diese Ergebnisse zur automatischen Bahnfolge untermauern die Bedeutung des systematischen Entwurfsprozesses und der statistischen Analyse des Lernerfolges neuronaler Netzwerke, nicht nur für die regelungstechnische Anwendung.

Der darauf folgende Vergleich von rein neuronaler Regelung und neuronal gestützter dynamischer Inversion zeigt, dass letztere eine höhere Bahnführungsgenauigkeit und bessere Lagestabilisierung aufweist. Trotz der Annahme eines zunächst idealisierten Inversionsmodells unterstreichen die Ergebnisse bereits die Vorzüge der Kombination aus modellbasierter und lernfähiger Regelung. Dabei belegt die durchgehende Verwendung des Gleitzustandslernverfahrens, dass die vom Regler selbsttätig durchgeführte Bestimmung der Lernrate in stabiler Weise geschieht.

Zuletzt wird die Adaptionfähigkeit des Gleitzustandsverfahrens in Gegenwart von Parameterunsicherheiten und Systemschäden untersucht und mit dem Rückpropagationsalgorithmus verglichen. Neben der Reaktion auf vorhandene Modellunsicherheiten werden die Antworten des Regelkreises auf sprungförmig auftretende Inversionsfehler analysiert. Dies geschieht sowohl für den Idealfall als auch für ein Flugzeug unter erheblichem Störeinfluss und zeigt, dass das gewählte Lernverfahren auftretende Fehler in jeder Situation rasch auszugleichen vermag. Die gleichzeitige Analyse der berechneten Lernraten und der Gewichtsnorm belegen eindrucksvoll, dass die Stabilitätsforderungen des Gleitzustandslernverfahrens auch in einem störungsbehafteten Regelkreis Gültigkeit besitzen. Außerdem kann auch für den geschlossenen Regelkreis gezeigt werden, dass eine zu hohe Wahl der Lernrate des Rückpropagationsverfahrens eine Destabilisierung zur Folge haben kann. Die daraus abzuleitende konservative Auslegung der festen Lernrate führt zu einer deutlichen Reduktion der Adaptionfähigkeit in Extremsituationen. Dieses wird an einem sowohl in Längs- wie auch Seitenbewegung statisch destabilisierten Flugzeug untersucht, welches ohne Stützung durch neuronale Netze nicht stabil geregelt werden kann. Dabei zeigt sich, dass das Lernverfahren mit Gleitzustand die Regelungsaufgabe bei nur geringem Qualitätsverlust bewältigen kann, während eine feste Lernrate bei zunehmend konservativer Auslegung starke Leistungseinbußen zu verzeichnen hat.

Zusammenfassend kann festgestellt werden, dass die Lernfähigkeit neuronaler Netzwerke ein nicht unerhebliches Potential zur Verbesserung existierender Regelungsverfahren besonders für vollautomatische und autonome Systeme besitzt. Durch den in der bisherigen Forschung weitgehend vernachlässigten Ansatz, neuronale Netze als geregelte Systeme zu betrachten, können regelungstechnisch motivierte Lernmethoden abgeleitet werden, die den existierenden Standardverfahren überlegen sind. Dies fällt besonders bei der automatischen Bestimmung der Lernrate ins Auge, da hierdurch empirische und stark idealisierte Auslegungsverfahren umgangen werden können. Ebenso ist es möglich, die geregelten Zustandsgrößen eines neuronalen Netzes systematisch in Phasendiagrammen zu analysieren, was den Vorwurf, es handle sich um eine *black box*, schwer haltbar macht.

Darüber hinaus zeigt sich, dass die Nutzung eines reproduzierbaren Entwurfsprozesses für jedwede Anwendung neuronaler Netze einen erheblichen ingenieurtechnischen Gewinn darstellt, da hierdurch bereits im Vorfeld eine systematische Analyse der Abbildungseigenschaften erfol-

gen kann. Die Ergebnisse der vorliegenden Arbeit belegen zudem, dass besonders in schwerwiegenden Extremsituationen ein System durch neuronale Netzwerke stabil regelbar bleibt, welches ohne diese Unterstützung vollends destabilisiert würde. Beide vorgestellten Regelungsverfahren untermauern außerdem die fundierten Abbildungsfähigkeiten neuronaler Netzwerke, auch komplexer nichtlinearer und stark verkoppelter dynamischer Zusammenhänge. Abschließend verbleibt festzustellen, dass besonders aus der Verknüpfung dieser Abbildungseigenschaften mit klassischer mathematisch-physikalischer Modellkenntnis ein regelungstechnischer Mehrwert zu erzielen ist, der einen Beitrag zur eingangs formulierten Problemstellung autonomen Verhaltens technischer Systeme leistet.

8.1 Zukünftige Forschungsansätze

Aus der Gesamtschau der Ergebnisse der vorliegenden Arbeit lassen sich eine Reihe weiterer aussichtsreicher wissenschaftlicher Ansatzpunkte ableiten. Diese betreffen sowohl die vorgestellten lernfähigen Regelungsverfahren und ihre Flugerprobung als auch substantielle Erweiterungen des Lernverfahrens basierend auf der Gleitzustandsregelung. Ihnen ist gemein, dass sie eine verstärkte interdisziplinäre Betrachtung von Regelungstechnik und maschinellem Lernen erfordern.

Der dargelegten Problematik, dass die modellfreie neuronale Regelung Zustandsgrößen des Systems zu begrenzten Schwingungen anregen kann, ließe sich mit einer Erweiterung der Reglerarchitektur entgegenwirken. Dazu könnte zum Beispiel auf das Konzept der Komitee-Entscheidungen neuronaler Netzwerke zurückgegriffen werden. Dabei werden die Ausgaben mehrerer Netzwerke nach variabel gestaltbaren Regeln zu einem konsolidierten Gesamtsignal synthetisiert. Dies ist sowohl für die Regler- als auch für die Beobachternetzwerke denkbar und bietet die Möglichkeit, derzeit nicht berücksichtigte Fehlerinformationen im Kritekelement des Regelkreises gezielt zu berücksichtigen. Eine derart verbesserte und gänzlich auf Maschinenlernverfahren beruhende Architektur ist besonders für die Regelung nur ungenau identifizierter Systeme geeignet.

Für die neuronal gestützte dynamische Inversion besteht ein wichtiger nächster Entwicklungsschritt in der Erprobung im Flugversuch. Besonders für unbemannte Kleinflugzeuge wäre es dabei wünschenswert, wenn auf die derzeit für die Regelung notwendige wie auch komplexe Windmessung verzichtet werden könnte. Dies ist unter Inkaufnahme von Zuständen in der internen Dynamik mit Verwendung der Eulerwinkel zur Lageregelung möglich. Neben einer genauen Stabilitätsbetrachtung eben dieser internen Dynamik wäre hierbei zu untersuchen, inwieweit erweiterte Flugzustandsbegrenzungen in die Architektur der Referenzmodelle einbezogen werden können, so dass deren Signale ungewollte Anregungen von Zuständen der internen Dynamik unterbinden. Ebenso sollte der in dieser Arbeit begonnene Ansatz der automatischen Generierung von Bordrechnercode komplexer Algorithmen zur beschleunigten Flugerprobung konsequent fortgeführt werden. Bei der Regelung der Fehlerdynamik besteht

darüber hinaus die Möglichkeit, die konventionellen Linearregler zum Beispiel durch Gleitzustandsregler zu ersetzen, womit die Robustheit des nicht neuronal gestützten Systems deutlich erhöht werden könnte.

Besonderes Entwicklungspotential ist für die erweiterte Anwendung der Gleitzustandsregelung auf neuronale Lernverfahren zu konstatieren. Die erfolgreiche Übertragung dieses Regelungsansatzes auf die Dynamik neuronaler Netze eröffnet die Möglichkeit einer vertieften Betrachtung ihrer Eigenschaften. Bisher sind neuronale Netzwerke vereinfacht als System zweiter Ordnung aufgefasst worden, was wie geschildert zu Schaltfunktionen erster Ordnung führt. Mit der Kenntnis, dass auch Gleitzustände höherer Ordnung herbeigeführt werden können, ist ebenfalls eine weniger vereinfachende Betrachtung neuronaler Netze möglich. Die Nutzung von Schaltfunktionen zweiter Ordnung etwa, also die Betrachtung eines Netzwerkes mit einer Stellgröße als nichtlineares System dritter Ordnung, würde bei der Regelung der Netzzustände durch die Vorgabe von Eigenfrequenz und Dämpfung gänzlich verändertes dynamisches Verhalten erlauben. Darüber hinaus ließe sich auch das angesprochene Konzept des *reaching law* auf neuronale Netze übertragen. Damit wird eine weitaus gezieltere Beeinflussung der Zustandstrajektorien im Phasenraum möglich, was dazu führt, dass die Eintreffphase in den Gleitzustand eine direkte Regelung erfährt und so die Konvergenzgeschwindigkeit des Lernverfahrens weiter erhöht wird. Besonders für die Verknüpfung von neuronalen Netzen und dynamischer Inversion ist dies von großer Bedeutung, da so auch die Fehlerdynamik von Mehrgrößensystemen eines höheren relativen Grades durch Netzwerke im Gleitzustand stabilisiert werden kann. Mit dieser erweiterten Betrachtungsweise können verbesserte, regelungstechnisch motivierte generische Lernverfahren entworfen werden, die auf Grund der variablen Lernrate und der Regelung der Netzwerkzustände den gängigen Ansätzen besonders in der Echtzeitanwendung überlegen sind.

Literaturverzeichnis

- [1] ROJAS, R.: *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin, 1996
- [2] HAYKIN, S.: *Neural Networks - A Comprehensive Foundation*. 2nd. Prentice Hall, Upper Saddle River, NJ, 1999
- [3] KRUSE, R. ; BORGELT, C. ; KLAWONN, F. ; MOEWES, C. ; RUSS, G. ; STEINBRECHER, M.: *Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Vieweg und Teubner Verlag, Wiesbaden, 2011
- [4] HORNIK, K. ; STINCHCOMBE, M. ; WHITE, H.: Multilayer Feedforward Networks are Universal Approximators. In: *Neural Networks 2* (1989), S. 359–366
- [5] FUNAHASHI, K. I.: On the Approximate Realization of Continuous Mappings by Neural Networks. In: *Neural Networks 2* (1989), S. 183–192
- [6] CYBENKO, G.: Approximation by Superpositions of a Sigmoidal Function. In: *Mathematics of Control, Signals and Systems 2* (1989), S. 303–314
- [7] CRONE, S. F.: *Neuronale Netze zur Prognose und Disposition im Handel*, Universität Hamburg, Diss., 2008
- [8] CRONE, S. F. ; KOURENTZES, N.: The impact of preprocessing on forecasting electrical load: An empirical evaluation of segmenting time series into subseries. In: *Proc. Int Neural Networks (IJCNN) Joint Conf*, 2011, S. 3285–3292
- [9] HEINERT, M.: *Systemanalyse der seismisch bedingten Kinematik Islands*, Institut für Geodäsie und Photogrammetrie, Technische Universität Braunschweig, Diss., 2008
- [10] VANDEGRIFT, J. ; WAGSTA, K. ; HO, G. ; PLAUGER, J.: Forecasting space weather: Predicting interplanetary shocks using neural networks. In: *Advances in Space Research* 36 (2005), S. 2323–2327
- [11] CHELLA, A. ; CIARLINI, P. ; MANISCALCO, U.: Neural Networks as Soft Sensors: a Comparison in a Real World Application. In: S (Hrsg.): *Proc. Int. Joint Conf. Neural Networks IJCNN '06*, 2006, S. 2662–2668

- [12] KAYGISIZ, B. ; ERKMEN, A. ; ERKMEN, I.: Enhancing positioning accuracy of GPS/INS system during GPS outages utilizing artificial neural network. In: *Neural Processing Letters* 25 (2007), S. 171–186
- [13] CHIANG, K. ; NOURELDIN, A. ; EL-SHEIMY, N.: Constructive neural-networks-based MEMS/GPS integration scheme. In: *IEEE Transactions on Aerospace and Electronic Systems* 44 (2008), Nr. 2, S. 582–594. <http://dx.doi.org/10.1109/TAES.2008.4560208>. – DOI 10.1109/TAES.2008.4560208
- [14] OMATU, S. ; KHALID, M. ; YUSOF, R.: *Neuro-Control and its applications - Advances in industrial control*. Berlin : Springer-Verlag, 1996
- [15] NARENDRA, K.: Neural networks for control theory and practice. In: *Proceedings of the IEEE* 84 (1996), Nr. 10, S. 1385–1406. <http://dx.doi.org/10.1109/5.537106>. – DOI 10.1109/5.537106
- [16] OMIDVAR, O. ; ELLIOTT, D.: *Neural Systems for Control*. Academic Press, 1997
- [17] SCHUMANN, J. ; LIU, Y.: *Applications of Neural Networks in High Assurance Systems*. ISBN 978-3-642-10689-7 : Springer Verlag, Berlin, 2010
- [18] HAGAN, M. T. ; MENHAJ, M. B.: Training Feedforward Networks with the Marquardt Algorithm. In: *IEEE Transactions on Neural Networks* 5 (1994), Nr. 6, S. 989–993. <http://dx.doi.org/10.1109/72.329697>. – DOI 10.1109/72.329697
- [19] CHEN, Y. ; ZHU, Q. ; LI, Z.: A Novel Artificial Neural Network Based on Hybrid PSO-BP Algorithm in the Application of Adaptive PMD Compensation System. In: *Lecture Notes in Computer Science* 4493 (2007), S. 311–319
- [20] DACHWALD, B.: *Low-Thrust Trajectory Optimisation and Interplanetary Mission Analysis Using Evolutionary Neurocontrol*, Universität der Bundeswehr München, Fakultät für Fakultät für Luft- und Raumfahrttechnik, Diss., 2003
- [21] STANLEY, K.: *Efficient Evolution of Neural Networks through Complexification*, Artificial Intelligence Laboratory, University of Texas, Austin, Diss., 2004
- [22] ALIPPI, C.: Selecting accurate, robust, and minimal feedforward neural networks. In: *IEEE Transactions on Circuits and Systems* 49 (2002), Nr. 12, S. 1799–1810. <http://dx.doi.org/.805710>. – DOI .805710
- [23] TRENN, S.: Multilayer Perceptrons: Approximation Order and Necessary Number of Hidden Units. In: *IEEE Transactions on Neural Networks* 19 (2008), Nr. 5, S. 836–844. <http://dx.doi.org/10.1109/TNN.2007.912306>. – DOI 10.1109/TNN.2007.912306

- [24] FERRARI, S. ; JENSENIUS, M.: A Constrained Optimization Approach to Preserving Prior Knowledge During Incremental Training. In: *IEEE Transactions on Neural Networks* 19 (2008), Nr. 6, S. 996–1009. <http://dx.doi.org/10.1109/TNN.2007.915108>. – DOI 10.1109/TNN.2007.915108
- [25] KAYNAK, O. ; ERBATUR, K. ; ERTUGRUL, M.: The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control - A Survey. In: *IEEE Transactions on Industrial Electronics* 48 (2001), Nr. 1, S. 4–17
- [26] NIED, A. ; SELEME, S. I. ; PARMA, G. G. ; MENEZES, B. R.: On-line neural training algorithm with sliding mode control and adaptive learning rate. In: *Neurocomputing* 70 (2007), S. 2687–2691
- [27] WANG, X. ; HUANG, Y.: Convergence Study in Extended Kalman Filter-Based Training of Recurrent Neural Networks. In: *IEEE Transactions on Neural Networks* 22 (2011), Nr. 4, S. 588–600. <http://dx.doi.org/10.1109/TNN.2011.2109737>. – DOI 10.1109/TNN.2011.2109737
- [28] BROCKHAUS, R. ; ALLES, W. ; LUCKNER, R.: *Flugregelung*. Springer Verlag, Berlin, 2011. – ISBN 978-3-642-01442-0
- [29] SEIFERT, J.: *Identifizierung nichtlinearer aerodynamischer Derivative mit einem Modulen Neuronalen Netzwerk*, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Institut für Systemdynamik und Flugmechanik, Diss., 2003
- [30] DAS, S. ; KUTTIERI, R. A. ; SINHA, M. ; JATEGAONKARJ, R.: Neural Partial Differential Method for Extracting Aerodynamic Derivatives from Flight Data. In: *Journal of Guidance, Control and Dynamics* 33 (2010), Nr. 2, S. 376–384
- [31] CALISE, A. ; YANG, B.-J. ; CRAIG, J. I.: Augmentation of an existing linear controller with an adaptive element. In: *Proc. American Control Conf the 2002* Bd. 2, 2002, S. 1549–1554
- [32] MENHAJ, M. B. ; RAY, S.: Neuro-based adaptive controller for longitudinal flight control. In: *Proc. Intelligent Control. 2003 IEEE Int. Symp*, 2003, S. 158–163
- [33] PASHILKAR, A.A. ; SUNDARARAJAN, N. ; SARATCHANDRAN, P.: A fault-tolerant neural aided controller for aircraft auto-landing. In: *Aerospace Science and Technology* 10 (2006), S. 49–61
- [34] PALLIX, J. ; DORAIS, G. ; PENIX, J.: Overview of Intelligent Systems & Operations Development / NASA. 2003. – Forschungsbericht
- [35] SCHUMANN, J. ; GUPTA, P. ; NELSON, S.: On Verification & Validation of Neural Network Based Controllers / NASA. 2003. – Forschungsbericht

- [36] JACKLIN, S. ; SCHUMANN, J. ; GUPTA, P. ; RICHARD, M. ; GUENTHER, K. ; SOARES, F.: Development of Advanced Verification and Validation Procedures and Tools for the Certification of Learning Systems in Aerospace Applications / NASA. 2005. – Forschungsbericht
- [37] ISIDORI, A.: *Nonlinear Control Systems*. Springer Verlag, Berlin, 1996
- [38] ADAMY, J.: *Nichtlineare Regelungen*. Springer Verlag, Berlin, 2009
- [39] LEWIS, F. L. ; YEGILDIREK, A. ; LIU, K.: Multilayer neural-net robot controller with guaranteed tracking performance. In: *IEEE Transactions on Neural Networks* 7 (1996), Nr. 2, S. 388–399. <http://dx.doi.org/10.1109/72.485674>. – DOI 10.1109/72.485674
- [40] CALISE, A.: Neural networks in nonlinear aircraft flight control. In: *IEEE Aerospace and Electronic Systems Magazine* 11 (1996), Nr. 7, S. 5–10. <http://dx.doi.org/10.1109/62.533965>. – DOI 10.1109/62.533965
- [41] CALISE, A. ; RYSDYK, R. T.: Nonlinear adaptive flight control using neural networks. In: *IEEE Control Systems Magazine* 18 (1998), Nr. 6, S. 14–25. <http://dx.doi.org/10.1109/37.736008>. – DOI 10.1109/37.736008
- [42] CALISE, A. ; LEE, H. ; KIM, N.: High Bandwidth Adaptive Flight Control. In: *AIAA Guidance, Navigation, and Control Conference*. Denver, CO, September 2000. – AIAA 2000-4551
- [43] JOHNSON, E. N.: *Limited Authority Adaptive Flight Control*, Georgia Institute of Technology, Diss., 2000
- [44] JOHNSON, E. N. ; CALISE, A. ; CORBAN, J. E.: Adaptive guidance and control for autonomous launch vehicles. In: *Proceedings of the IEEE Aerospace Conference* Bd. 6, 2001, S. 2669–2682
- [45] KIM, N.: *Improved Methods in Neural Network-Based Adaptive Output Feedback Control with Applications to Flight Control*, School of Aerospace Engineering, Georgia Institute of Technology, Diss., 2003
- [46] YERRAMALLA, S. ; FULLER, E. ; MLADENOVSKI, M. ; CUKIC, B.: Lyapunov Analysis of Neural Network Stability in an Adaptive Flight Control System. In: *Self-Stabilizing Systems* Bd. 2704, 2003 (Lecture Notes in Computer Science), S. 77–92
- [47] RYSDYK, R. ; CALISE, A.: Robust Nonlinear Adaptive Flight Control for Consistent Handling Qualities. In: *IEEE Transactions on Control Systems Technology* 13 (2005), Nr. 6, S. 896–910
- [48] KANNAN, S.: *Adaptive Control of Systems in Cascade with Saturation*, Georgia Institute of Technology, Diss., 2005

- [49] BURKEN, J. ; NGUYEN, N. T. ; GRIFFIN, B.: Adaptive Flight Control Design with Optimal Control Modification on an F-18 Aircraft Model. In: *AIAA Infotech@Aerospace Conference*. Atlanta, Georgia, April 2010. – AIAA 2010-3364
- [50] ZHANG, X. ; LIU, Y. ; RYSDYK, R. ; KWAN, C. ; XU, R.: An intelligent hierarchical approach to actuator fault diagnosis and accommodation. In: *Proc. IEEE Aerospace Conference*, 2006
- [51] NGUYEN, N. ; KRISHNAKUMAR, K. ; KANESHIGE, J. ; NESPECA, P.: Dynamics and Adaptive Control for Stability Recovery of Damaged Asymmetric Aircraft. In: *AIAA Guidance, Navigation and Control Conference*. 21 - 24 August 2006, Keystone, Colorado, 2006
- [52] HOLZAPFEL, F.: *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*, Lehrstuhl für Flugmechanik und Flugregelung, Technische Universität München, Diss., 2004
- [53] NECSULESCU, D. ; JIANG, Y.-W. ; KIM, B.: Neural Network Based Feedback Linearization Control of an Unmanned Aerial Vehicle. In: *International Journal of Automation and Computing* 1 (2007), Nr. 4, S. 71–79
- [54] NOWACK, J.: *Windkanal-Freiflugmessungen zur Bestimmung flugmechanischer Kenngrößen*, Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen, Diss., 2010
- [55] LORENZ, S.: *Adaptive Regelung zur Flugbereichserweiterung des Technologiedemonstrators ARTIS*, Technische Universität Braunschweig, Diss., 2010
- [56] MCCULLOCH, W. ; PITTS, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. In: *Bulletin of Mathematical Biophysics* 5 (1943), S. 115–133
- [57] HEBB, D.: *The Organization of Behavior*. John Wiley, New York, 1949
- [58] ROSENBLATT, F.: The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. In: *Psychological Review* 65 (1958), S. 386–408
- [59] RUSSELL, S. ; NORVIG, P.: *Künstliche Intelligenz - Ein moderner Ansatz*. Pearson Education, München, 2004. – ISBN 3-8273-7089-2
- [60] IZHIKEVICH, E. M.: *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press, Cambridge, Massachusetts, London, England, 2007
- [61] FERRARI, S. ; STENGEL, R.F: Online Adaptive Critic Flight Control. In: *Journal of Guidance, Control and Dynamics* 27 (2004), Nr. 5, S. 777–786
- [62] BRONSTEIN, I. N. ; SEMENDJAJEW, K. A. ; MUSIOL, G. ; MÜHLIG, H.: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2001

- [63] KROGMANN, U.: *Beitrag zur Anwendung Neuronaler Netze in der Flugregelung*, Fakultät für Maschinenbau und Elektrotechnik, Technische Universität Braunschweig, Diss., 1995
- [64] LEVIN, A. U. ; NARENDRA, K. S.: Control of nonlinear dynamical systems using neural networks. II. Observability, identification, and control. In: *IEEE Transactions on Neural Networks* 7 (1996), Nr. 1, S. 30–42. <http://dx.doi.org/10.1109/72.478390>. – DOI 10.1109/72.478390
- [65] HEINERT, M.: Artificial neural networks - how to open the black boxes? In: *Application of Artificial Intelligence in Engineering Geodesy*. Wien, Österreich, 2008, S. 42–62. – ISBN 978-3-9501492-4-1
- [66] MACKALL, D. ; NELSON, S. ; SCHUMMAN, J.: *Verification & Validation Of Neural Networks For Aerospace Systems / NASA*. 2002. – Forschungsbericht
- [67] PARMA, G. G. ; MENEZES, B. R. ; BRAGA, A. P.: Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks. In: *IEEE Electronics Letters* 34 (1998), Nr. 1, S. 97–98
- [68] YU, X. ; KAYNAK, O.: Sliding-Mode Control With Soft Computing: A Survey. In: *IEEE Transactions on Industrial Electronics* 56 (2009), Nr. 9, S. 3275–3285
- [69] UTKIN, V. ; YU, X. (Hrsg.) ; XU, J.-X. (Hrsg.): *Lecture Notes in Control and Information Sciences*. Bd. 274: *Variable Structure Systems: Towards the 21st Century*. First Stage of VSS: People and Events. Springer Verlag, Berlin, 2002
- [70] UTKIN, V.: *Sliding Modes in Control and Optimization*. Springer Verlag, Berlin, 1992
- [71] TRUONG, T.: *Parametervariable und strukturvariable Regelkonzepte für elektrische Antriebssysteme mit mechanisch-elastischer Übertragung*, Fakultät Elektrotechnik und Informationstechnik, Technische Universität Chemnitz, Diss., 2002
- [72] FÖLLINGER, O.: *Methoden der Regelungs- und Automatisierungstechnik - Nichtlineare Regelungen I*. Oldenbourg Verlag, München, 7. Auflage, 1993
- [73] HUNG, J. ; GAO, W. ; HUNG, J.: Variable Structure Control: A Survey. In: *IEEE Transactions on Industrial Electronics* 40 (1993), Nr. 1, S. 2–22
- [74] UTKIN, V. ; GULDNER, J. ; SHI, J.: *Sliding Mode Control in Electro-Mechanical Systems*. CRC Press, London, 2009
- [75] GAO, W. ; HUNG, J.: Variable Structure Control of Nonlinear Systems: A New Approach. In: *IEEE Transactions on Industrial Electronics* 40 (1993), Nr. 1, S. 45–55
- [76] MALKIN, J.: *Theorie der Stabilität einer Bewegung*. Oldenbourg Verlag, München, 1959

- [77] LASALLE, J. ; LEFSCHETZ, S.: *Die Stabilitätstheorie von Ljapunow*. Bibliographisches Institut, Mannheim, 1967
- [78] UNBEHAUEN, H.: *Regelungstechnik II - Zustandsregelungen, digitale und nichtlineare Regelsysteme*. Vieweg Verlag, Braunschweig, 1989. – ISBN 3-528-43348-5
- [79] MÖSSNER, M.: *Optimierung einer adaptiven neuronalen Reglerstruktur und Stabilitätsanalyse der verwendeten Lernverfahren*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2009
- [80] KRÜGER, T. ; MÖSSNER, M. ; KUHN, A. ; AXMANN, J. ; VÖRSMANN, P.: Adaptive Flight Control for Unmanned Aircraft Using a Stable Neural Network Observer. In: *AIAA Infotech@Aerospace 2010*. Atlanta, Georgia, USA, 20 - 22 April 2010. – AIAA-2010-3396
- [81] KRÜGER, T. ; SCHNETTER, P. ; PLACZEK, R. ; VÖRSMANN, P.: Nonlinear Adaptive Flight Control Using Sliding Mode Online Learning. In: *International Joint Conference on Neural Networks*. San José, California, USA, July 31 - August 5 2011, S. 2997–2904
- [82] SCHOLZ, G.: *Modellfreie adaptive Flugregelung mittels neuronaler Netzwerke*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2011
- [83] SACHS, F.: *Analyse der Rekonfigurationseigenschaften eines Flugreglers auf Basis dynamischer Inversion*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2011
- [84] EFE, M. ; KAYNAK, O. ; X. YU ; YU, X. (Hrsg.) ; XU, J.-X. (Hrsg.): *Lecture Notes in Control and Information Sciences*. Bd. 274: *Variable Structure Systems: Towards the 21st Century*. Variable Structure Systems Theory in Computational Intelligence. Springer Verlag, Berlin, 2002
- [85] NIED, A. ; SELEME, S. I. ; PARMA, G. G. ; MENEZES, B. R.: On-line Adaptive Neural Training Algorithm for an Induction Motor Flux Observer. In: *IEEE Power Electronic Specialists Conference - PESC2005* (2005)
- [86] PARMA, G. G. ; MENEZES, B. R. ; BRAGA, A. P.: Improving backpropagation with sliding mode control. In: *Proc. Vth Brazilian Symp. Neural Networks*, 1998, S. 8–13
- [87] PARMA, G. G. ; MENEZES, B. R. ; BRAGA, A. P.: Sliding mode backpropagation: control theory applied to neural network learning. In: *Proc. Int. Joint Conf. Neural Networks IJCNN '99* Bd. 3, 1999, S. 1774–1778
- [88] EFE, M. O. ; KAYNAK, Okyay: Stabilizing and Robustifying the Learning Mechanisms of Artificial Neural Networks in Control Engineering Applications. In: *International Journal of Intelligent Systems* 15 (2000), S. 365–388

- [89] TOPALOV, A. V. ; KAYNAK, O.: Online Learning in Adaptive Neurocontrol Schemes with a Sliding Mode Algorithm. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 31 (2001), Nr. 3, S. 445–450
- [90] SHAKEV, N. G. ; TOPALOV, A. V. ; KAYNAK, O.: Sliding Mode Algorithm for Online Learning in Analog Multilayer Feedforward Neural Networks. In: *LNCS* 2714 (2003), S. 1064–1072
- [91] SCHECK, M.: *Anwendung von stabilen Sliding-Mode Trainingsverfahren auf neuronale Regler*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2011
- [92] SARPTURK, S. ; ISTEANOPULOS, Y. ; KAYNAK, O.: On the Stability of Discrete-Time Sliding Mode Control Systems. In: *IEEE Transactions on Automatic Control* AC-32 (1987), October, Nr. 10, S. 930–932
- [93] BRÜNING, G. ; HAFFER, X.: *Flugleistungen - Grundlagen, Flugzustände, Flugabschnitte*. Springer Verlag, Berlin, 1978. – ISBN 978-3-642-01442-0
- [94] *Deutsche Luftfahrtnorm LN 9300: Begriffe, Größen und Formelzeichen in der Flugmechanik*. Beuth Verlag, Berlin, 1970
- [95] KRÜGER, T.: *Erweitertes Kalman-Filter zur Sensordatenfusion in Navigationsanwendungen*. Diplomarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2011
- [96] GALL, C.: *Stützung der Lagebestimmung mittels Mehrempfänger-GPS*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2010
- [97] WINKLER, S.: *Zur Sensordatenfusion für integrierte Navigationssysteme unbemannter Kleinstflugzeuge*, Zentrum für Luft- und Raumfahrt, Technische Universität Braunschweig, Diss., 2007
- [98] SCHULZ, H.-W.: *Aufbau und Durchführung von Versuchen zur Vermessung von Modellbau-Servos und Ermittlung theoretischer Modelle zur Beschreibung des dynamischen Übertragungsverhaltens*. Diplomarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2002
- [99] FÖLLINGER, O.: *Regelungstechnik - Einführung in die Methoden und ihre Anwendung*. Hüthig Buch Verlag Heidelberg, 1992
- [100] ELVERS, S.: *Entwurf eines selbstlernenden neuronalen Reglers für unbemannte Fluggeräte*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2009

- [101] ZIGANN, P.: *Optimierung eines Echtzeitbetriebssystems für Mikroautopiloten zur Implementierung neuronaler Netze*. Masterarbeit, Institut für Luft- und Raumfahrtssysteme und Institut für wissenschaftliches Rechnen, Technische Universität Braunschweig, 2011
- [102] KRÜGER, A.: *Entwurf und Test einer adaptiven neuronalen Reglerstruktur für unbemannte Fluggeräte*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2008
- [103] KRÜGER, T. ; KRÜGER, A. ; KUHN, A. ; AXMANN, J. ; VÖRSMANN, P.: Adaptive Flight Control for Unmanned Aerial Vehicles Using a Neural Network Predictor. In: *Deutscher Luft- und Raumfahrtkongress*. Aachen, September 2009. – ausgewählte Manuskripte, ISBN:978-3-932182-63-4, PaperID 121206
- [104] KRÜGER, T. ; MÖSSNER, M. ; KUHN, A. ; AXMANN, J. ; VÖRSMANN, P.: Sliding Mode Online Learning for Flight Control Applications in Unmanned Aerial Systems. In: *WCCI - World Congress on Computational Intelligence*. Barcelona, Spain, July 2010. – pp. 3738-3745
- [105] SCHNETTER, P.: *Entwicklung eines Flugreglers auf Basis dynamischer Inversion*. Diplomarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2011
- [106] PLACZEK, R.: *Analyse und Optimierung neuronaler Elemente in adaptiven Regelungsverfahren*. Studienarbeit, Institut für Luft- und Raumfahrtssysteme, Technische Universität Braunschweig, 2010
- [107] KRÜGER, T. ; SCHNETTER, P. ; VÖRSMANN, P.: Nichtlineare, lernfähige Flugregelung am Beispiel eines unbemannten Fluggerätes mit Systemschäden. In: *Deutscher Luft- und Raumfahrtkongress*. Bremen, September 2011. – ISBN:978-3-932182-68-5, PaperID 161314
- [108] KRÜGER, T. ; SCHNETTER, P. ; PLACZEK, R. ; VÖRSMANN, P.: Fault-Tolerant Nonlinear Adaptive Flight Control Using Sliding Mode Online Learning. In: *Neural Networks* 32 (2012), S. 267–274
- [109] UNBEHAUEN, H.: *Regelungstechnik I - Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme*. Vieweg Verlag, Braunschweig, 1992. – ISBN 3-528-63332-8
- [110] NOLLAU, R.: *Modellierung und Simulation technischer Systeme*. Springer Verlag, Berlin, 2009. – ISBN 978-3540891208
- [111] SCHNIEDER, E.: *Methoden der Automatisierung - Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme*. Vieweg Verlag, Braunschweig, 1999

- [112] NARENDRA, K. ; PARTHASARATHY, K.: Identification and control of dynamical systems using neural networks. In: *IEEE Transactions on Neural Networks* 1 (1990), Nr. 1, S. 4–27. <http://dx.doi.org/10.1109/72.80202>. – DOI 10.1109/72.80202
- [113] SCHULZ, H.-W.: *Ein rekonfigurierbares Bildverarbeitungssystem für unbemannte Kleinflugzeuge - Entwicklung eines bildgestützten Landeverfahrens*, Zentrum für Luft- und Raumfahrt, Technische Universität Braunschweig, Diss., 2008
- [114] PARK, J.-B.: *A Contribution to the Development of Sense and Avoid Systems for Autonomous Flight Missions*, Campus Forschungsflughafen, Technische Universität Braunschweig, Diss., 2010
- [115] KRÜGER, T. ; WILKENS, C.-S. ; REINHOLD, M. ; SELSAM, P. ; BÖHM, B. ; VÖRSMANN, P.: Ergebnisse des ANDROMEDA-Projektes - Automatische Luftbildgewinnung mit Unbemannten Kleinflugzeugen. In: *Deutscher Luft- und Raumfahrtkongress*. Hamburg, September 2010. – ISBN:978-3-932182-68-5, PaperID 161314
- [116] CALISE, A. ; LEE, S. ; SHARMA, M.: Development of a Reconfigurable Flight Control Law for the X-36 Tailles Fighter Aircraft. In: *AIAA Guidance, Navigation, and Control Conference*. Denver, CO, September 2000. – AIAA 2000-3940
- [117] FU, L.-C.: Neural Network Approach to Variable Structure Based Adaptive Tracking of SISO Systems. In: *IEEE Workshop on Variable Structure Systems*, 1996, S. 148–153
- [118] DIERKS, T. ; JAGANNATHAN, S.: Output Feedback Control of a Quadrotor UAV Using Neural Networks. In: *IEEE Transactions on Neural Networks* 21 (2010), Nr. 1, S. 50–66
- [119] NARENDRA, K. ; ANNASWAMY, A.: A New Adaptive Law for Robust Adaptation without Persistent Excitation. In: *IEEE Transactions on Automatic Control* 32 (1987), Nr. 2, S. 134–145. <http://dx.doi.org/10.1109/TAC.1987.1104543>. – DOI 10.1109/TAC.1987.1104543
- [120] BARTOLINI, G. ; LEVANT, A. ; PISANO, A. ; USAI, E. ; YU, X. (Hrsg.) ; XU, J.-X. (Hrsg.): *Lecture Notes in Control and Information Sciences*. Bd. 274: *Variable Structure Systems: Towards the 21st Century*. Higher-Order Sliding Modes for the Output-Feedback Control of Nonlinear Uncertain Systems. Springer Verlag, Berlin, 2002
- [121] KROONENBERG, A.: *Airborne Measurement of Small-Scale Turbulence with special regard to the Polar Boundary Layer*, Zentrum für Luft- und Raumfahrt, Technische Universität Braunschweig, Diss., 2009
- [122] POLYCARPOU, M. ; TRUNOV, A.: Learning Approach to Nonlinear Fault Diagnosis Detectability Analysis. In: *IEEE Transactions on Automatic Control* 45 (2000), Nr. 4, S. 806–812

A Anhang

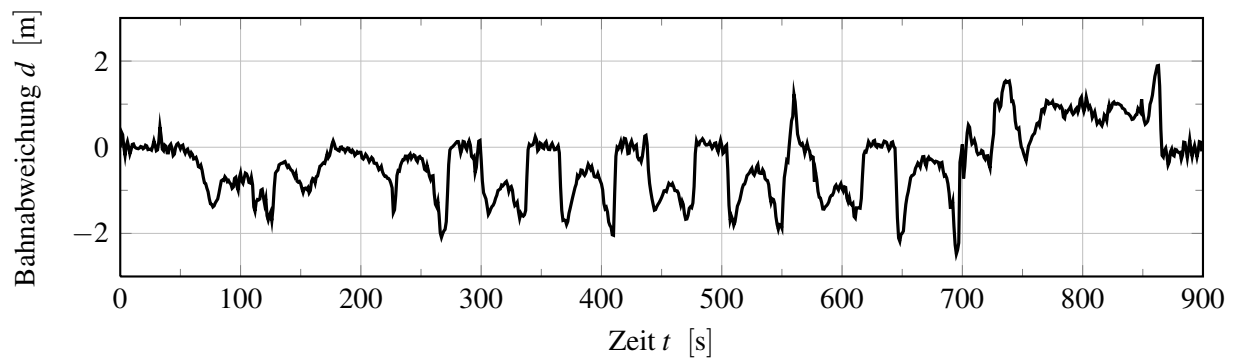


Abbildung A.1: Seitliche Bahnabweichung der Beispielmmission mit dynamischer Inversion.

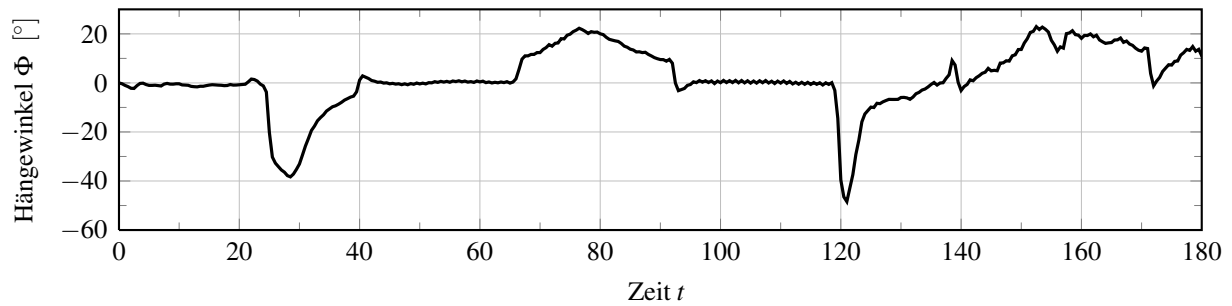


Abbildung A.2: Hängewinkelverlauf des *P360* unter Einfluss atmosphärischer Störungen.

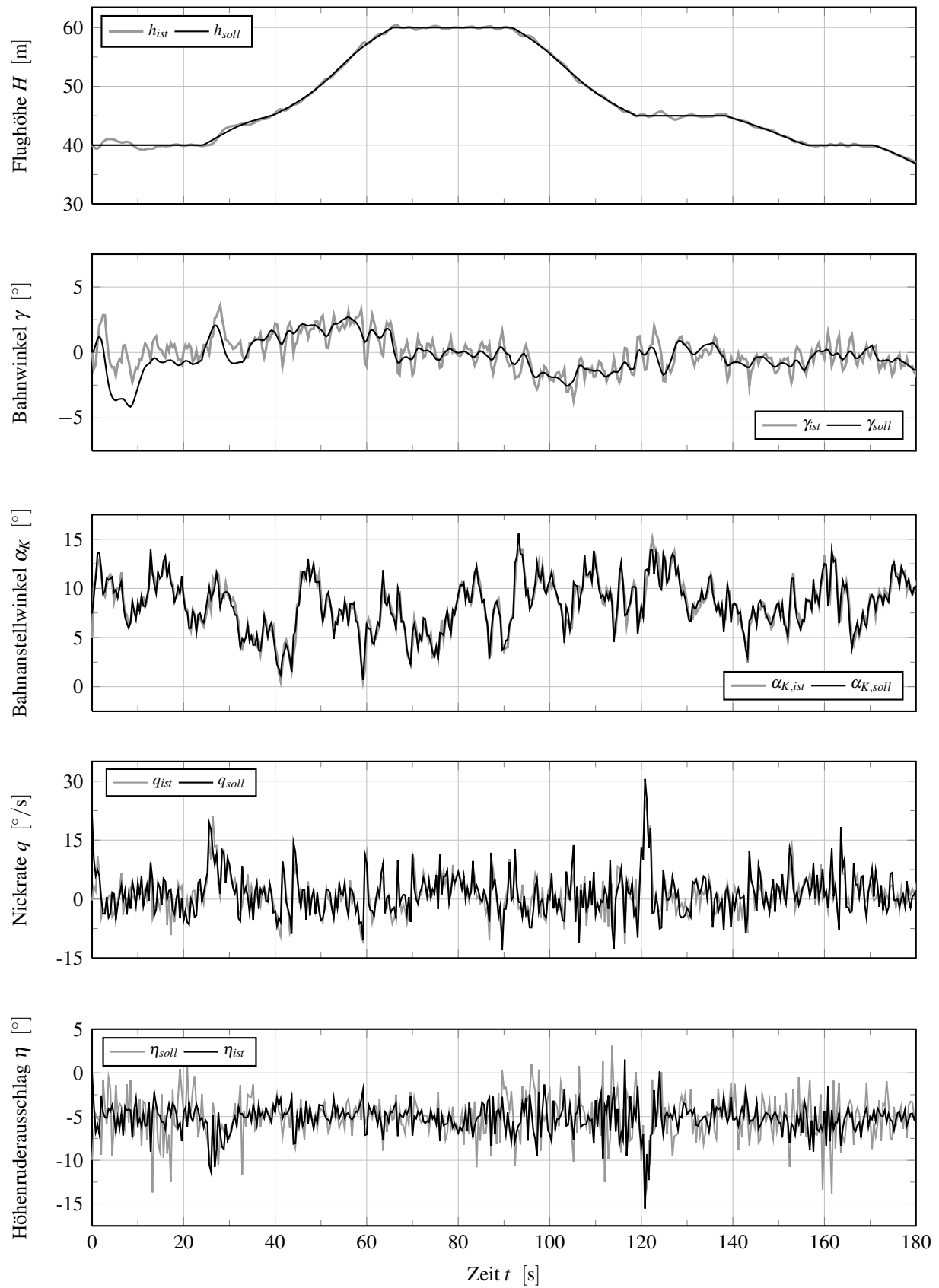


Abbildung A.3: Größen der Längsbewegung unter Windeinfluss mit Modellfehler von -10 %.

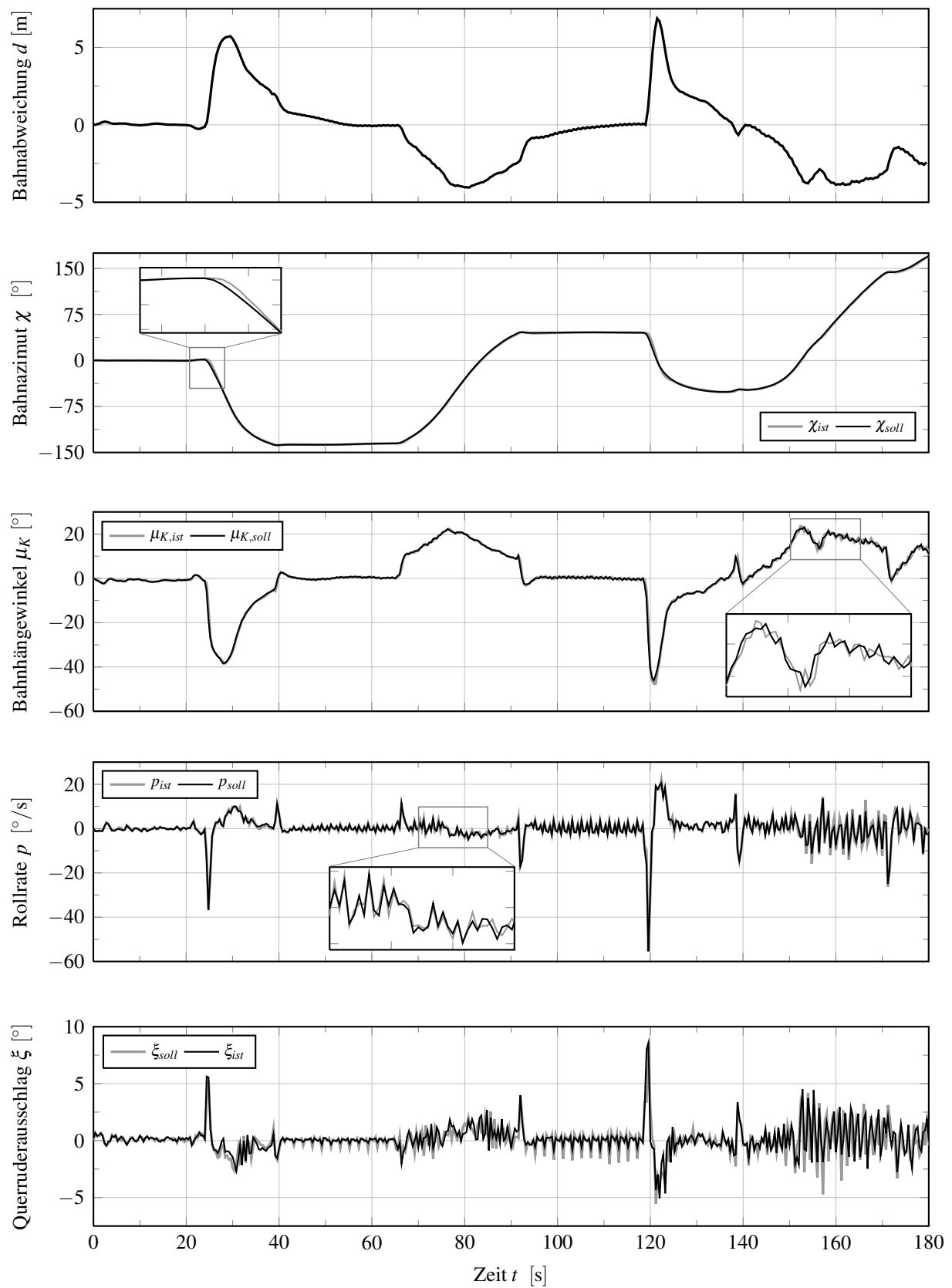


Abbildung A.4: Größen der Seitenbewegung unter Windeinfluss mit Modellfehler von -10 %.

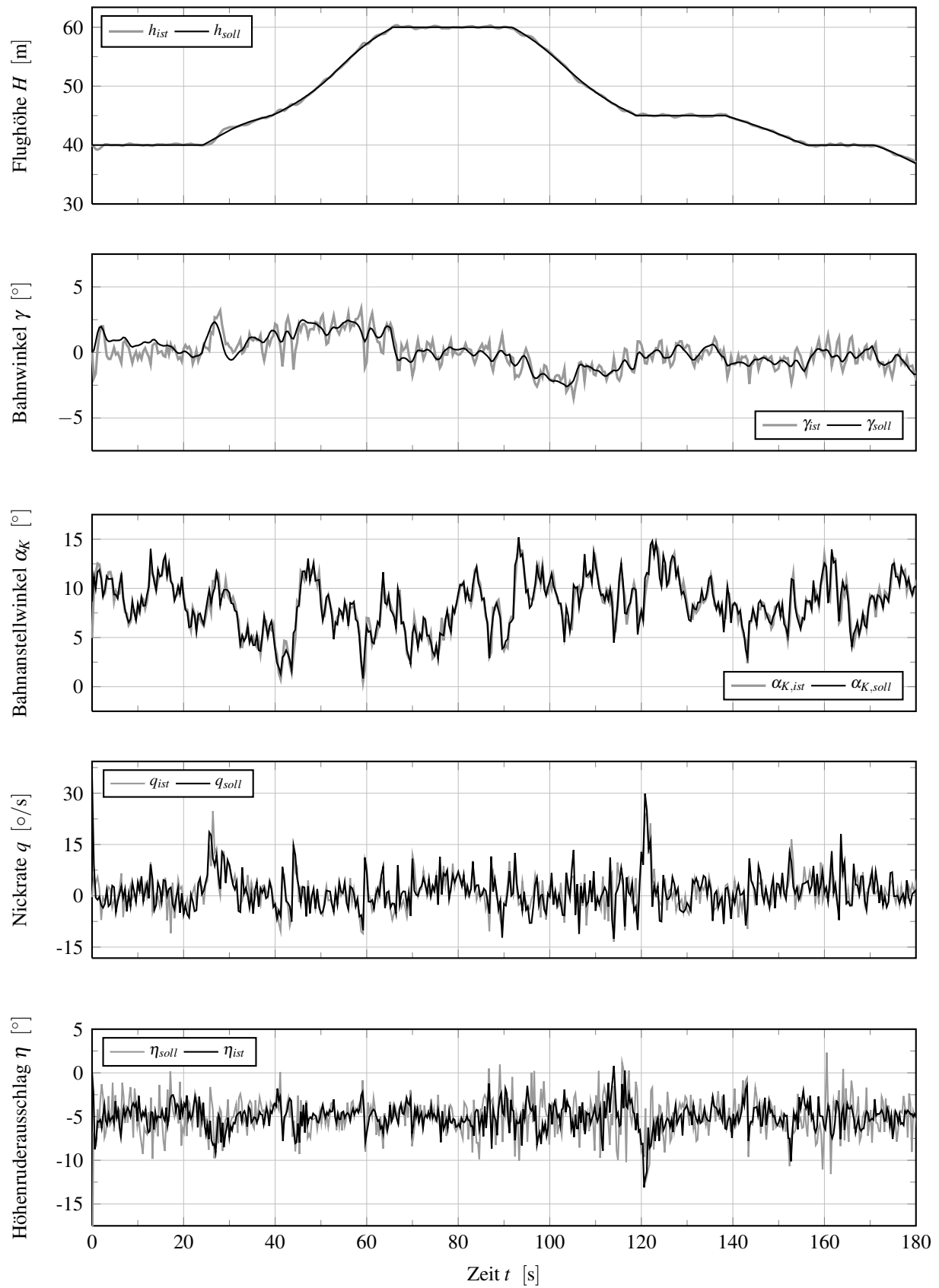


Abbildung A.5: Größen der Längsbewegung unter Windeinfluss mit Modellfehler von +10 %.

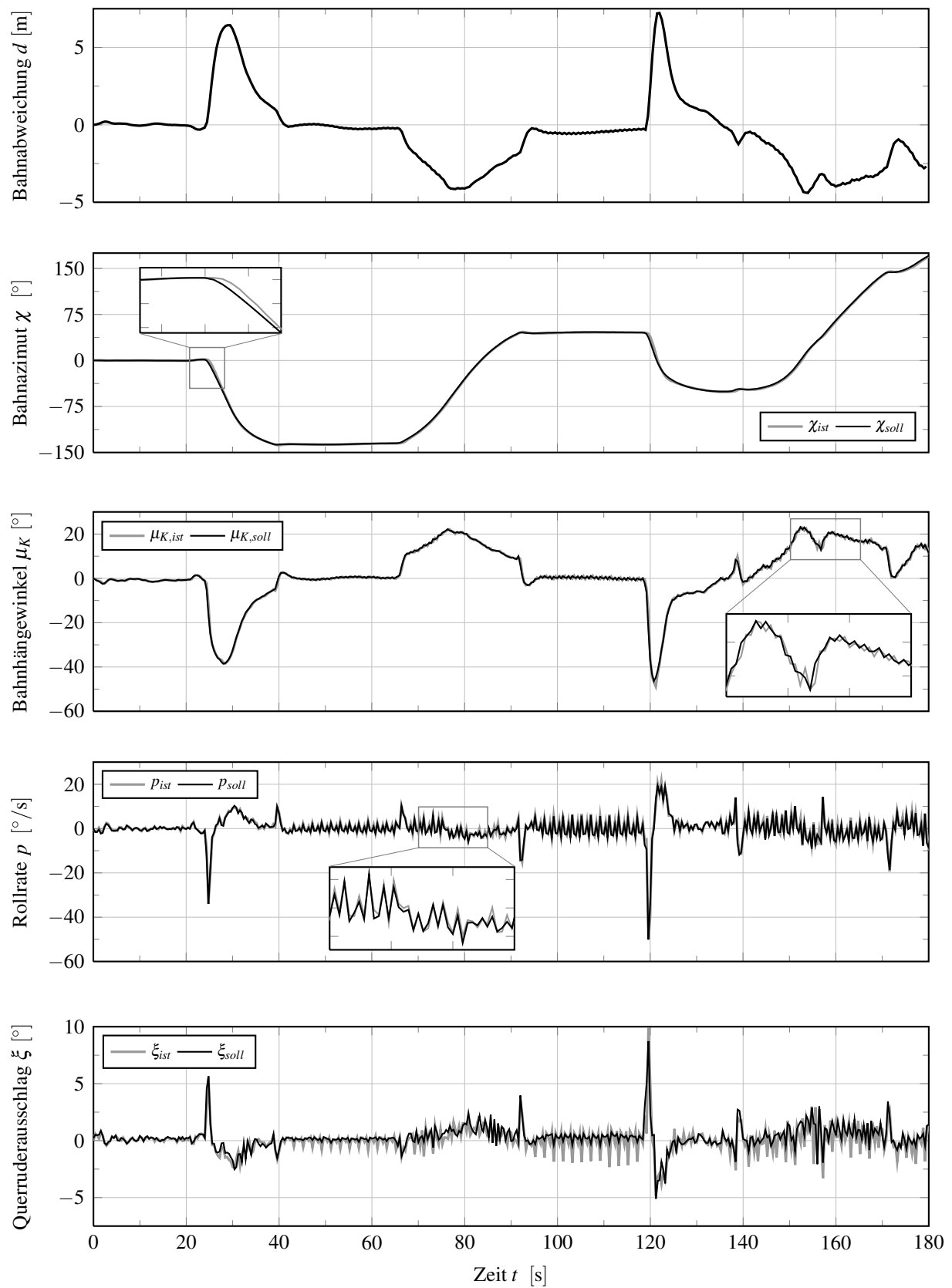


Abbildung A.6: Größen der Seitenbewegung unter Windeinfluss mit Modellfehler von +10 %.

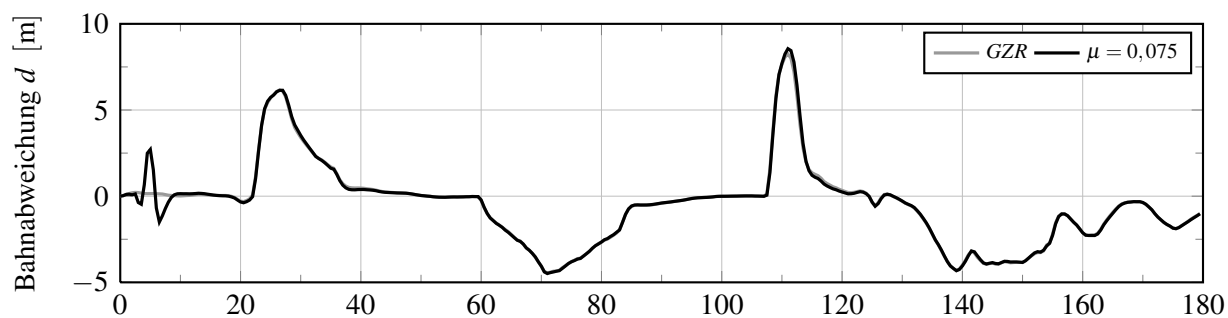


Abbildung A.7: Vergleich der seitlichen Bahnabweichung der verschiedenen Lernverfahren.

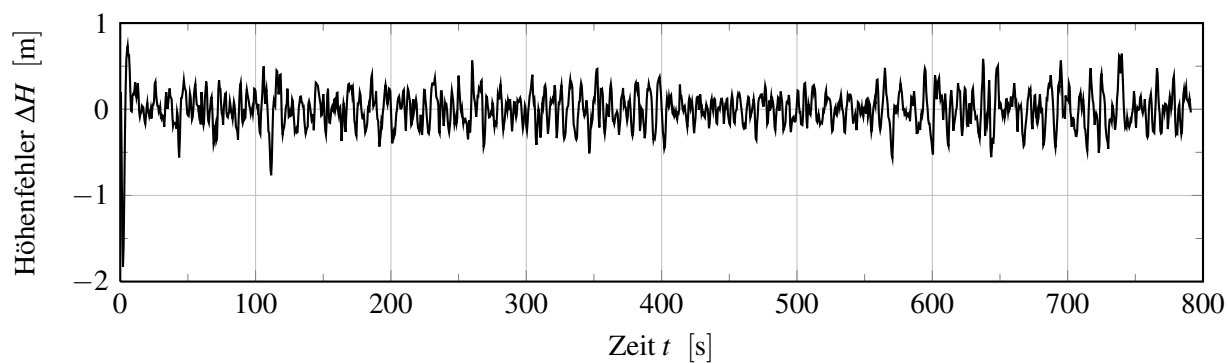


Abbildung A.8: Höhenfehler des *P360* beim Flug mit Systemschaden.

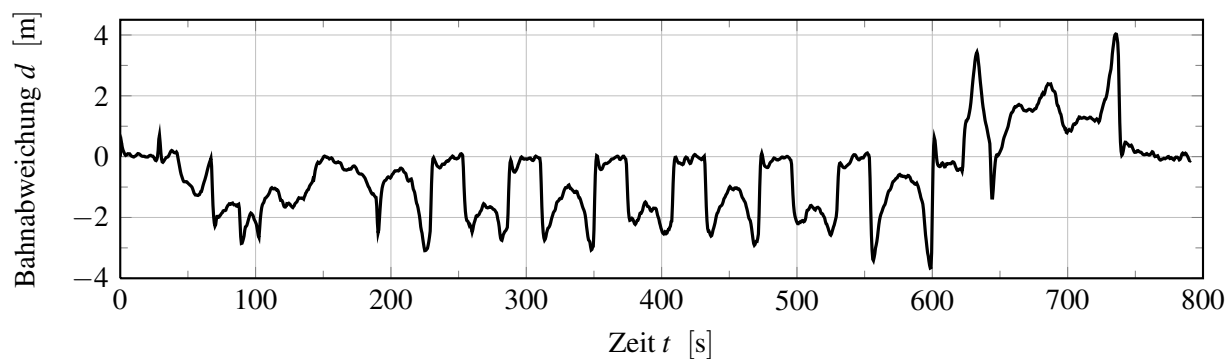


Abbildung A.9: Seitliche Bahnabweichung des *P360* beim Flug mit Systemschaden.